MICROCONTROLLORI Componenti 3D Visione artificiale SENSORI POLVERI Progammazione PIC Pascal tastiera PS2 PIC MikroBASIC Raspberry Commodore64 Arduino etilometro EFFETTO BLANKING tutorial PIC18

fare elettronica
www.farelettronica.com
327 SETTEMBRE Mensile Anno XXVIII € 6,00

**Tutorial**
**Programmare i PIC in Pascal**

**Sensori di polveri fini**

**Gestione tastiera PS/2 con PIC**

**Componenti in 3D con Proteus**

**Modifichiamo il vecchio Commodore 64**

# RASPBERRY PI
## *Il computer in una mano*

## Etilometro con Arduino

MOUSER ELECTRONICS
Distribuzione
di semiconduttori
e componenti per i
progettisti elettronici
Distributore Autorizzato
mouser.com

SEI ABBONATO? COMPRI LA RIVISTA IN EDICOLA?
DA OGGI PUOI SCARICARE O ACQUISTARE

elektor

in formato PDF!

Tramite il nuovo,
esclusivo servizio

Adobe PDF

IE cloud

www.ie-cloud.it

Veloce, sempre puntuale
e sempre disponibile sul tuo PC.

mouser.com
Distribuzione di semiconduttori e componenti elettronici per ingegneri e progettisti.

Distributore autorizzato

# Da noi il servizio clienti non ha barriere linguistiche

Quando si tratta di fornire le tecnologie piú avanzate in tempi brevi, possedere le necessarie competenze locali é fondamentale. Tramite la sua presenza mondiale, Mouser é in grado di fornire un supporto alle vendite commerciale e tecnico professionale e sempre a vostra disposizione.

**Italia**
Centro Direzionale, Milanofiori, Strada 1 Palazzo E1
20090 Assago-MI, Italia
+39 02 575065 71 | italy@mouser.com

Texas Instruments Authorized Distributor · Maxim · Fairchild Semiconductor · ON Semiconductor · Atmel · Infineon · OSRAM · Microchip · freescale · Vishay · Silicon Labs · Avago · NXP · CREE

CODICE MIP 2815395

mouser.com | Prodotti d'avanguardia per progetti innovativi

MOUSER ELECTRONICS
a tti company

Mouser e Mouser Electronics sono marchi commerciali registrati di Mouser Electronics, Inc. Tutti gli altri prodotti, loghi e denominazioni di società qui citati possono essere marchi commerciali dei rispettivi proprietari.

# sommario

**327** *settembre 2012*

## Zoom in


**24** **PC EMBEDDED: IL PROGETTO RASPBERRY PI (PARTE PRIMA)**

In un mondo in continuo cambiamento, anche il personal computer si evolve e diventa pocket, con il nuovo Raspberry Pi. In questo articolo introduttivo vediamo innanzitutto come il Raspberry Pi si pone nel mercato dei PC embedded.

*di Piero Boccadoro*


## Progettare & costruire


**34** **UN ETILOMETRO CON ARDUINO**

Realizziamo un semplice etilometro costituito da un sensore Mq-3 e la piattaforma Arduino. Uno strumento utile a evitare spiacevoli e pericolose situazioni.

*di Leonardo Neri*

**40** **SENSORI DI POLVERI FINI**

Sempre più spesso le amministrazioni comunali ricorrono al blocco del traffico automobilistico nelle grandi città (nonostante i dubbi sull'efficacia di tali misure di contrasto dell'inquinamento), segnale della crescente gravità dell'impatto del particolato atmosferico sulla nostra salute. Approfondiamo quindi i principi di funzionamento dei vari sensori elettronici esistenti per misurare in tempo reale la concentrazione e le dimensioni delle polveri presenti nell'aria.

*di Marco Carminati*

**44** **CORSO MIKROBASIC LA LIBRERIA PS/2**

In questo numero impareremo a utilizzare, in maniera approfondita, i display LCD testuali, soffermandoci sul modello fornito con la scheda di sviluppo EasyPIC.

*di Giovanni Di Maria*

**52** **MULTIKERNAL PER COMMODORE 64**

Nei periodi gloriosi degli 8 bit per gli utenti del C64 era pratica diffusa sostituire il kernal per ottenere prestazioni superiori. Oggi, grazie a due semplici circuiti e pochi componenti, è possibile selezionare ben otto diversi kernal con la semplice pressione di un tasto.

*di Massimo Barbagallo*



*Rispondi* e... **VINCI!** *pag.* **22**

DIVERTITI E METTI ALLA PROVA LE TUE CONOSCENZE CON **ELETTRO QUIZ** E VINCI OGNI MESE ESCLUSIVI PREMI!

# IMPARA A PROGRAMMARE ARDUINO con i nostri strumenti didattici

*Libri in italiano con esercitazioni didattiche e tutti i materiali necessari per imparare a programmare facilmente con la piattaforma Arduino. Particolarmente indicati per corsi scolastici e per quanti affrontano per la prima volta la programmazione di microcontrollori.*

## Libro "Primi passi con Arduino"

Pensato per i principianti che si avvicinano all'elettronica e per gli hobbisti che non hanno mai affrontato la programmazione. Con dieci esercitazioni didattiche.

cod. ARDUBOOK1
**€ 9,90**

## Libro "Primi passi con ARDUINO " + scheda Arduino UNO R3

cod. ARDUKITBOOK1
**€ 29,90**

## Libro "Primi passi con Arduino" + Starter Kit con Arduino UNO Rev3

cod. ARDUKITBOOK2
**€ 56,90**

Set composto dal libro "Primi passi con Arduino" e dallo starter kit cod. ARDUKITBOOK comprendente la scheda Arduino UNO Rev3 e tutto il materiale necessario per realizzare gli esperimenti didattici escritti nel libro.

## Starter kit con Arduino UNO R3

Comprende la scheda Arduino UNO R3 e tutto il materiale necessario per realizzare gli esperimenti didattici descritti nel volume "Primi passi con ARDUINO".

cod. ARDUKITBOOK
**€ 52,00**

## Arduino UNO Rev3 con ATMEGA328

Arduino UNO R3 è la più recente versione della scheda di sperimentazione basata su ATmega328.

cod. ARDUINOUNOREV3
**€ 24,50**

NOVITÀ

*Dopo il libro "base" ecco la versione "advanced"!*

solo **€ 22,00**

## "ARDUINO UNO Programmazione avanzata e Librerie di sistema"

Gli approfondimenti e le informazioni destinate a chi vuole utilizzare Arduino per progetti evoluti in un libro di facile consultazione e ricco di spunti. Con dieci esercitazioni didattiche avanzate.

cod. ARDUADVANCED

FUTURA ELETTRONICA
www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Caratteristiche tecniche di questi prodotti e acquisti on-line su *www.futurashop.it*

Prezzi IVA inclusa.

CODICE MIP 2814652

di MAURIZIO DEL CORSO

# Computer in TASCA

***Dopo il ciclone "Arduino" ecco affacciarsi sul mercato un nuovo prodotto che affascinerà sicuramente il pubblico di hobbisti e professionisti: Raspberry Pi.***

Nel corso degli anni abbiamo assistito a molte rivoluzioni tecnologiche: la miniaturizzazione si è spinta a livelli molto avanzati e ciò ha consentito lo sviluppo di microcontrollori sempre più performanti in package sempre più piccoli. Gli sviluppatori hanno colto questa opportunità, lanciando sul mercato prodotti che segneranno un'epoca nell'elettronica. Dopo Arduino, ecco che arriva Raspberry Pi, un computer di dimensioni ridottissime, ma dalle prestazioni veramente notevoli. Basato su Linux, può essere collegato direttamente a un TV tramite connettore RCA video-composito o HDMI. Sicuramente troverà molte applicazioni, soprattutto in campo multimediale. Visto che crediamo nelle potenzialità di questo nuovo ritrovato tecnologico, abbiamo pensato di dedicargli una serie di articoli che vi guideranno nella programmazione e nell'integrazione del Raspberry Pi, sfruttando al massimo tutte le sue caratteristiche. Ma le novità non finiscono qui. Iniziamo con questo numero un tutorial sulla programmazione dei PIC in Pascal, mentre nella sezione progetti troverete tutte le istruzioni per realizzare un etilometro con Arduino. Un nuovo numero ricco di informazioni tutto da gustare al rientro dalle vostre vacanze!

# FIERA DELL'ELETTRONICA

ELECTRONIC days

CODICE MIP 2815718

# MANTOVA

## presso PalaBam

# 27-28 OTTOBRE 2012

**orario continuato dalle 09.00 alle 18.00**

## FIERA DELL'ELETTRONICA IN GENERE

## + MERCATINO SURPLUS

MANIFESTAZIONI FIERISTICHE ITALFIERE S.R.L.

PALA BAM PALAMANTOVA S.R.L.

www.italfiere.net

**BUONO SCONTO DI € 1,00**

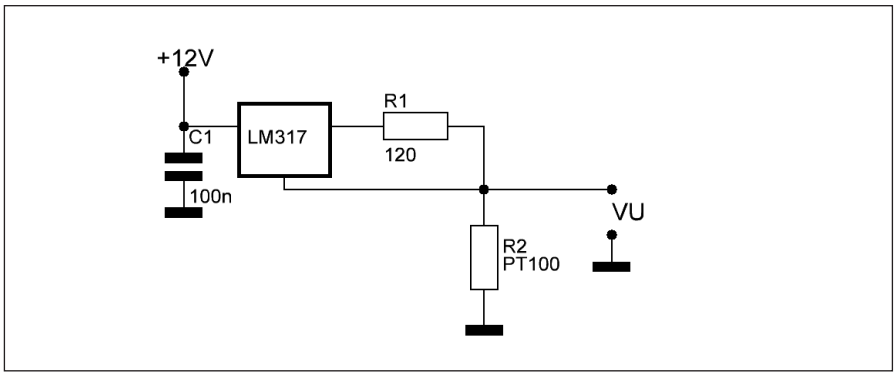
Da presentare alla biglietteria (non cumulabile)

*Fiera Elettronica MANTOVA 27-28 ottobre 2012*

**Per ricevere lo sconto alle prossime edizioni compila questo modulo:**

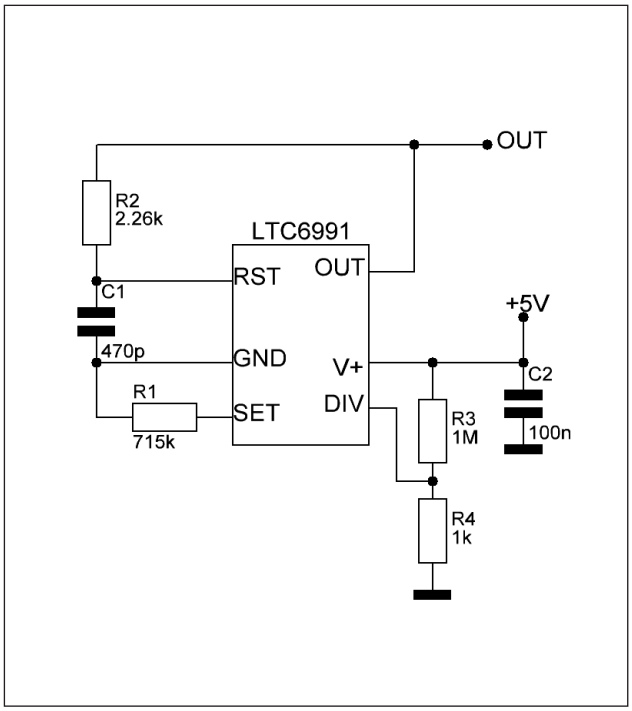
Email: ..... Provincia di provenienza: .....

Come hai saputo di questa fiera? .....




## CIRCUITO DI CONDIZIONAMENTO PT100

*Il PT100 è una termo resistenza PTC in grado di variare la propria resistenza interna al variare della temperatura. Trova largo utilizzo come dispositivo di sicurezza negli alimentatori che dissipano molto calore. In figura è riportato lo schema di un generatore di corrente costante sfruttando LM317 che, fornendo una corrente costante di 10 mA, alimenta il PT100 garantendo l'affidabilità dell'uscita.*

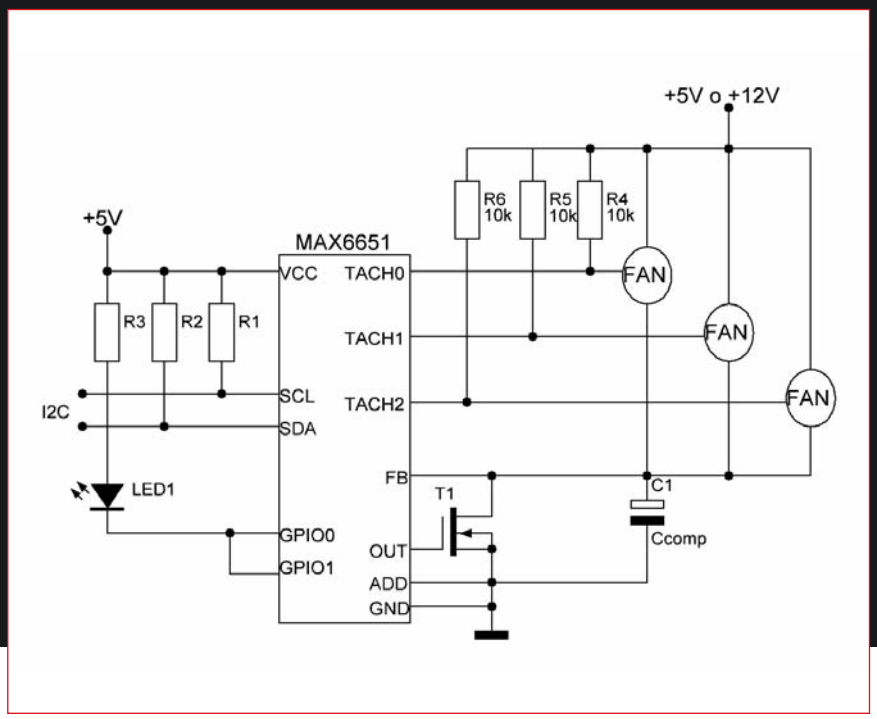


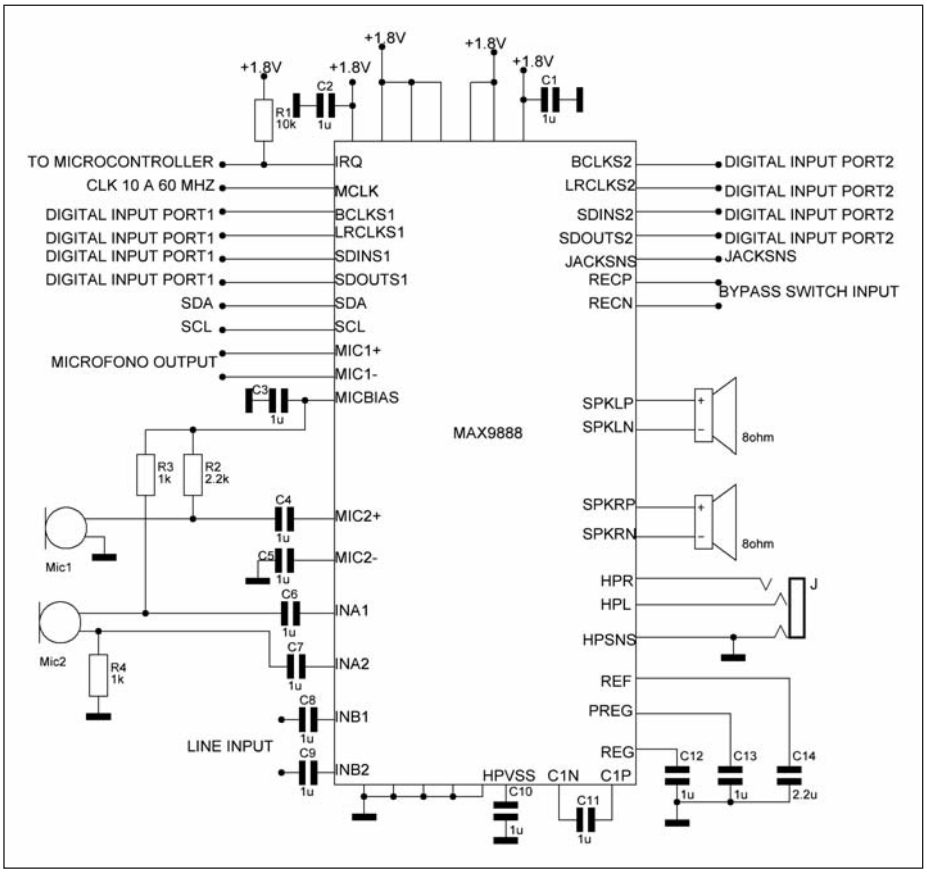

## OSCILLATORE PROGRAMMABILE

LTC6991 è un oscillatore al silicio programmabile in un range di periodo che va da 1.024 ms a 9,54 ore (29,1 Hz a 977 Hz), realizzato per essere inserito in applicazioni che richiedono molta precisione e che devo durare molto tempo. Viene alimentato con una tensione che va dai 2,2 V ai 5V.

## Controller per ventole

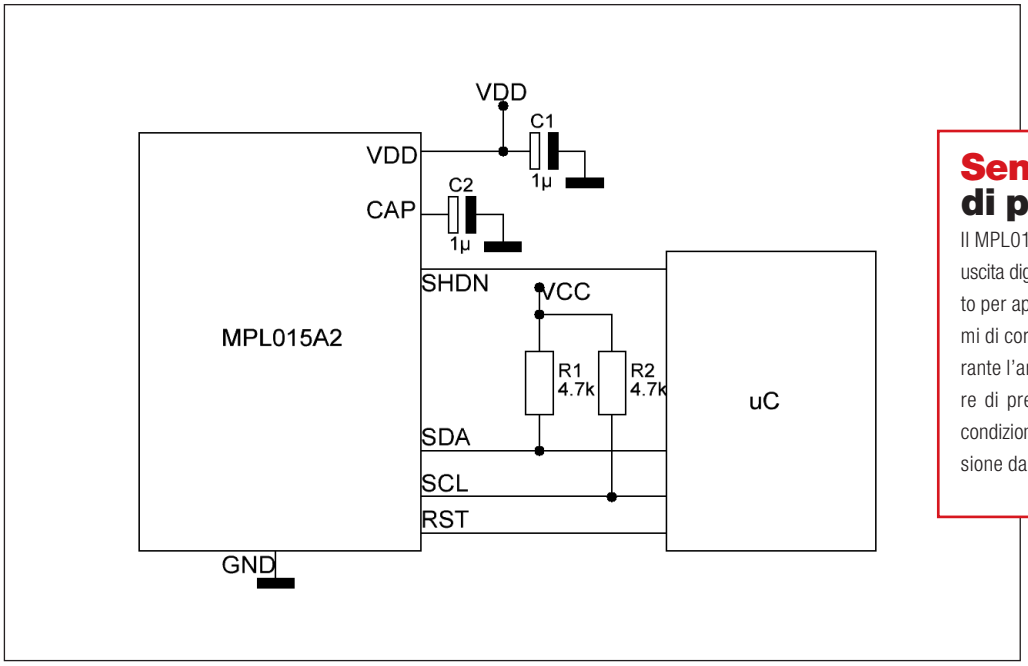
I controllori FAN MAX6650/MAX6651 utilizzano un'interfaccia I2C compatibile per regolare e monitorare la velocità delle ventole 5 V DC / 12 V DC brushless. Il MOSFET o transistor bipolari esterno serve per regolare linearmente la velocità delle ventole. MAX6650 regola la velocità di una singola ventola e ne monitora il suo funzionamento, mentre il MAX6651 può controllare fino a quattro ventole. Questi due integrati vengono alimentati con una tensione compresa tra 3 V e 5 V. La ventola viene alimentata con la sua tensione nominale totalmente indipendente dal circuito integrato.



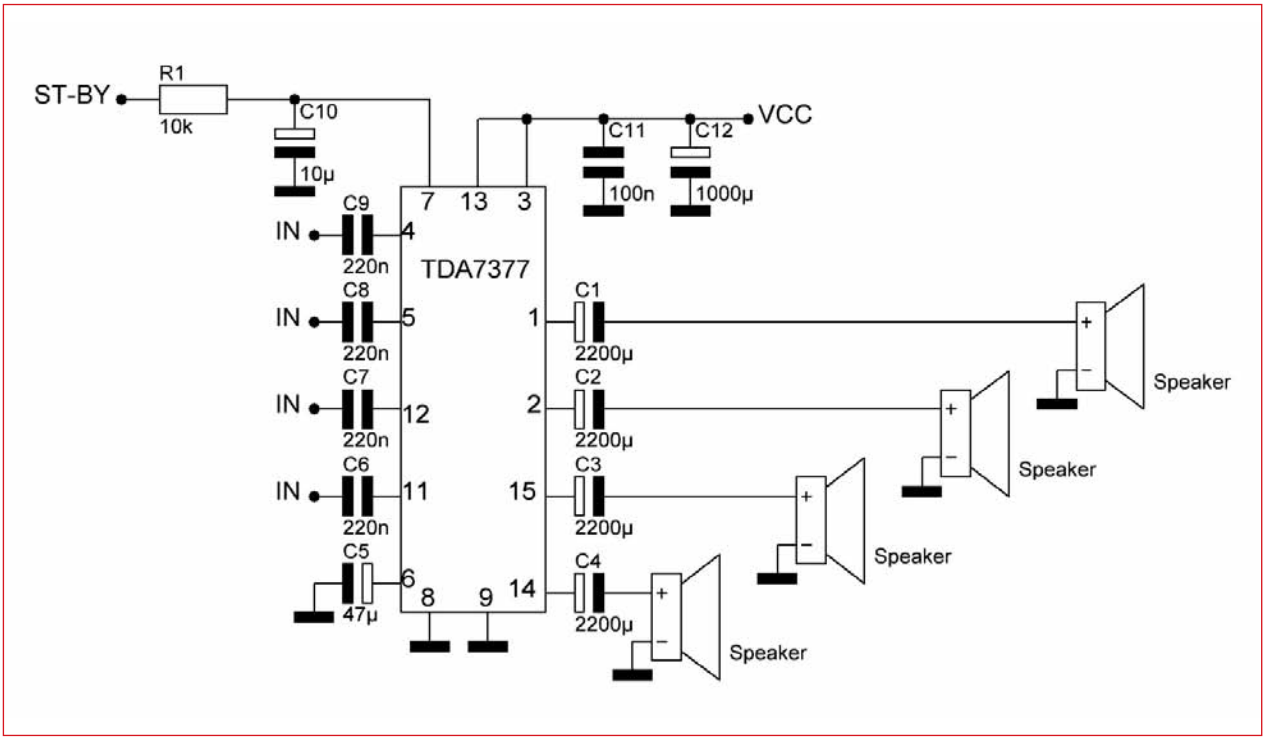

## CODEC Audio

**Il MAX9888 è un full-optional CODEC audio con elevate prestazioni e bassi consumi che lo rendono ideale per applicazioni portatili. Gli amplificatori interni di Classe D forniscono l'amplificazione sufficiente per i due altoparlanti. Viene alimentato con una tensione compresa tra 2,8 V e 5,5 V. Viene impiegato per applicazioni come MP3 e telefoni cellulari.**



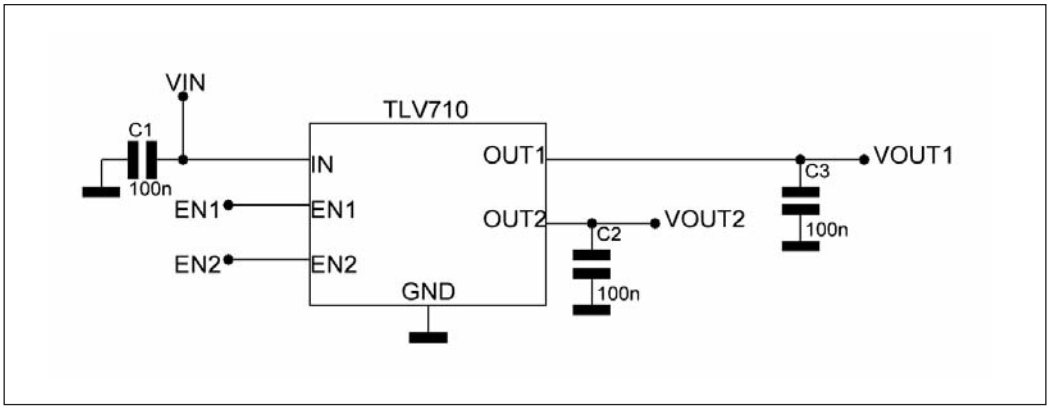

## Sensore di pressione

Il MPL015A2 è un sensore di pressione assoluta con uscita digitale a basso costo. Il suo Package LGA è adatto per applicazioni elettroniche portatili. Bassi consumi di corrente 5 A nella modalità Active e 0,06 A durante l'arresto (Sleep). MPL015A2 impiega un sensore di pressione MEMS con un circuito integrato di condizionamento per fornire la misurazione della pressione da 50-115 kPa.




## AMPLI IN CLASSE AB

*Il TDA7377 è un amplificatore per auto di classe AB in grado di funzionare in modalità MONO e in modalità stereo o a 4 altoparlanti. La sua potenza di uscita varia in base all'impedenza del diffusore sonoro, ad esempio riesce a fornire 30 W con due casse da 4 ohm con una tensione di alimentazione di 18 V. Viene fornito in un package Multiwatt15V e Multiwatt15H.*




## STABILIZZATORE A BASSA TENSIONE

*IL TLV710 è stato sviluppato per applicazioni portatili, avendo il vantaggio di stabilizzare la tensione in ingresso variabile da 0,3 a 6 V, fornendo in uscita una tensione di 2,8 V con 200 mA di corrente massima erogabile. Possiede un unico ingresso ed è possibile abilitare o disabilitare il canale di uscita attraverso due pin di controllo. Essendo studiato per applicazioni portatili non richiede molti componenti esterni.*

CODICE MIP 2826540

# Be the WaveAce

Il WaveAce esiste per far funzionare le cose.

Questo tipo di Tecnico è sveglio. I suoi strumenti di Misurazione sono leggendari. Lui semplifica i collaudi ed affronta la sua prossima sfida in un lampo.

Tu sei un WaveAce? Scoprilo su **bethewaveace.com.**

**Oscilloscopi WaveAce 1000 & 2000**

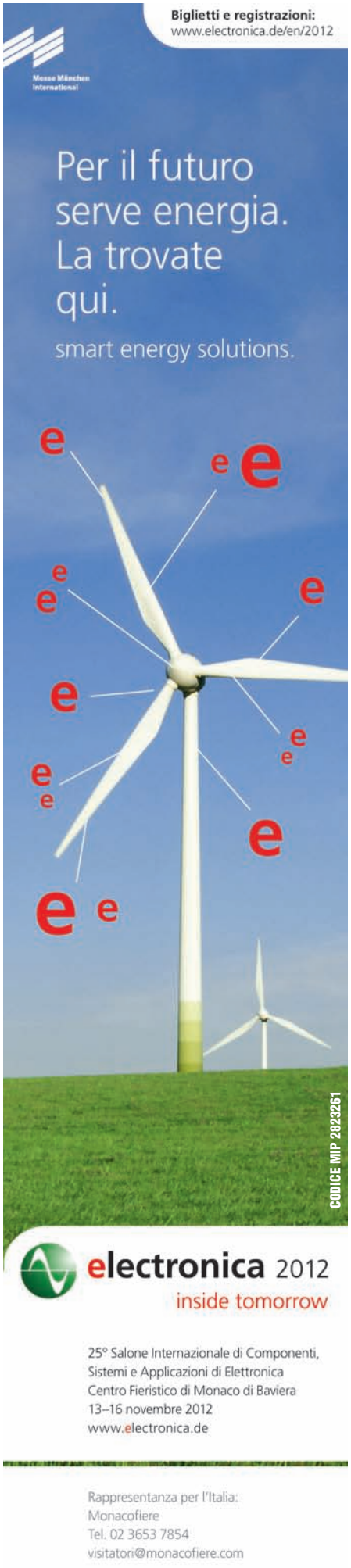
- Banda passante da 40 MHz a 300 MHz
- Campionamento fino a 2 GS/s
- Memoria fino a 2 Mpts
- A partire da 710 EUR

Disponibile presso:

Selint s.r.l.
Via del Fontanile Anagnino, 50
00118 ROMA
Tel 06.79841028
Fax 06.79845609
e-mail: info@selint.it
web: www.selint.it

AUTHORIZED LeCroy DISTRIBUTOR

LeCroy
Beyond the Limits

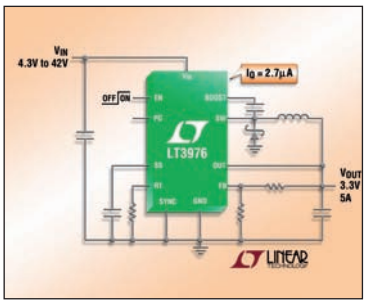


## RS Components e Allied Electronics consentono ora ordini illimitati di Raspberry Pi

RS Components e Allied Electronics sono ora in grado di gestire ordini illimitati per Raspberry Pi, il computer con le dimensioni di una carta di credito sviluppato per incentivare la diffusione della programmazione tra i giovani e gli appassionati. I clienti di RS e Allied di tutto il mondo possono ora ordinare quantità multiple della scheda Raspberry Pi Model B, insieme agli accessori coordinati, quali le card SD con installato il sistema operativo e i case per proteggere la scheda. Nel momento dell'ordine, a tutti i clienti verrà data una stima della data di consegna. Fino a oggi era possibile inoltrare il proprio ordine di Raspberry Pi solamente attraverso lo Store Pi online di RS e Allied, ora invece aziende, programmatori professionisti e istituti di istruzione possono effettuare un ordine attraverso le versioni locali dei siti di RS e Allied. I clienti intenzionati ad acquistare Raspberry Pi per scopo privato, invece, dovranno accedere al Pi Store per i loro acquisti. Mentre gli ordini in corso di Raspberry Pi verranno spediti entro i prossimi mesi, l'incremento della produzione ha consentito a RS e Allied di abolire la restrizione del numero di unità ordinabili da un unico cliente. Gli ordini possono ora essere effettuati per un numero illimitato di Raspberry Pi e accessori coordinati senza l'obbligo di registrazione o di ricevere un invito all'acquisto. Questi ordini verranno evasi a partire dalla fine di Settembre. Il prezzo di vendita di Raspberry Pi di RS e Allied è di 27,40 euro più IVA, spese di consegna e dazi doganali, ove applicati.

***CODICE MIP 2826550***

## Convertitori DC/DC step-down sincroni da 2MHz, 42V e 2,5A (IOUT) e 5A (IOUT) che richiedono una corrente di riposo di soli 2,7µA




***Linear Technology Corporation annuncia l'LT3975, un regolatore di commutazione step-down da 42 V in grado di fornire 2,5 A di corrente in uscita continua e richiede solo 2,7 µA di corrente di riposo. Allo stesso modo, l'LT3976 può funzionare da un ingresso da 40 V, fornisce una corrente di uscita di 5 A e richiede solo 3,3 µA di corrente di riposo. Entrambi i dispositivi offrono un intervallo di tensioni di ingresso nominali compreso tra 4,2 V e 40 V, per cui sono l'ideale per le applicazioni automotive e industriali. Gli switch interni da 75 mOhm offrono un rendimento fino al 90%. Il funzionamento Burst Mode dei dispositivi LT3975 e LT3976 offre una bassissima corrente di riposo. Il design esclusivo dei dispositivi fornisce una tensione di dropout di soli 500 mV quando la tensione di ingresso scende al di sotto della tensione programmata. L'utente può programmare la frequenza di commutazione da 200kHz a 2MHz e sincronizzarla tra 250 kHz e 2 MHz. L'LT3975 e l'LT3976 utilizzano uno switch altamente efficiente da 75 mOhm, dotato di diodo di boost, oscillatore e circuito logico e di controllo integrati in un unico die. Il funzionamento Burst Mode a basso ripple garantisce elevato rendimento con correnti di uscita ridotte, mantenendo il ripple di uscita al di sotto di 15 mVPK-PK.***

***CODICE MIP 2826598***

## Mitsubishi Electric presenta i nuovi videoproiettori multimedia data

Mitsubishi Electric presenta la nuova Serie 700 DLP 3D Ready di videoproiettori multimedia data. Disponibile in due modelli, WD720U e XD700U, la nuova Serie 700 offre prestazioni eccellenti in termini di qualità delle immagini, grazie a un maggior livello di contrasto e di luminosità, flessibilità e facilità d'uso, oltre ad un significativo risparmio energetico. L'elevata luminosità da 5000 lumen con un rapporto di contrasto di 3000:1 permette alla Serie 700 di riprodurre immagini in maniera chiara e precisa indipendentemente dalle caratteristiche del luogo di proiezione. Inoltre, la presenza dell'originale Mitsubishi Electric Color Wheel con tecnologia BrilliantColor incorporato permette una riproduzione del colore assolutamente reale, qualsiasi sia la fonte di connessione. La funzione Lens Shift, consente di settare manualmente la posizione verticale della lente, evitando di inclinare il videoproiettore per correggere la posizione dell'immagine, permettendo così una maggiore flessibilità d'installazione. I proiettori della Serie 700 DLP 3D Ready sono dotati di eccellenti funzioni audio che permettono la trasmissione simultanea di uno speaker e di una fonte audio esterna; tutto ciò è possibile anche grazie alla bassa emissione di rumore della ventola di raffreddamento.

***CODICE MIP 2826994***

## Molex amplia le opzioni di personalizzazione, modifica ed estensione della gamma Mini-Fit e del sistema KK

***Molex Incorporated offre una gamma ampliata di opzioni di personalizzazione, modifica ed estensione di alcune varianti dei suoi prodotti Mini-Fite del suo sistema KK di connettori "modulari" per applicazioni a passo ridotto, alta densità e correnti elevate. Le versioni CME delle interconnessioni Mini-Fit e KK sono dotate di una maggiore scelta di parametri di potenza, numero di circuiti, configurazioni e dimensioni, a garanzia di una flessibilità ottimale nelle applicazioni filo-filo, filo-scheda e scheda-scheda standard e con accoppiamento alla cieca per prodotti a media potenza. I prodotti di interconnessione Mini-Fit sono adatti per applicazioni di alimentazione fino a 13,0 A per circuito e sono disponibili nelle varianti a fila singola o doppia, con un numero di opzioni circuitali compreso fra 2 e 24 e con più alternative in termini di placcatura e materiali. La versatilità degli header e delle prese KK consente migliaia di configurazioni per applicazioni con passo compreso fra 2,50 e 5,08 mm, correnti fino a 13,0 A per circuito e tensioni di segnale fino a 600 V.***

***CODICE MIP 2827007***

## Modulo Bluetooth 2.0 + EDR in classe 2 con ottimo rapporto prezzo prestazioni

Sena Technologies, Inc. ha annunciato l'aggiunta alla serie di prodotti Parani (famosa linea di prodotti dedicati all'industrial Bluetooth) del nuovo Parani-BCD210: un modulo Bluetooth v2.0 + EDR Class 2 OEM SMD/DIP a basso consumo e dall'ottimo rapporto prezzo prestazioni. Il Parani-BCD210 con antenna è studiato per i produttori OEM che vogliono implementare la funzionalità Bluetooth in Classe 2 nei loro prodotti in modo tempestivo e con costi contenuti. Il BCD210 supporta il Bluetooth di classe 2 con trasmissione a basso livello di potenza e con la distanza di comunicazione fino a 300m in linea d'aria a vista. Il BCD210 supporta le interfacce UART, USB, I2C, PCM, PIO per la comunicazione con i prodotti OEM. Il BCD210 supporta il profilo SPP (Serial Port Profile) del Bluetooth e può gestire fino a quattro connessioni multiple simultanee. In alternativa, il BCD210 può anche essere fornito con profilo HCI per poter essere gestito da un host. È possibile anche sviluppare e integrare il proprio firmware nel BCD210. Il BCD210 è qualificato completamente con specifiche Bluetooth v2.0 + EDR in maniera tale che i produttori OEM possono risparmiare costi e tempi per le certificazioni dei propri prodotti.

***CODICE MIP 2827000***

CODICE MIP 2827306

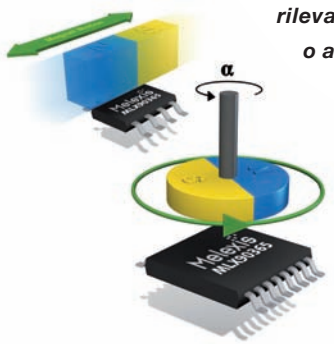
# LEM porta la misurazione di grande precisione della corrente ai pacchi batterie dei veicoli elettrici

LEM annuncia i trasduttori di corrente di elevata precisione CAB che consentono ai produttori di pacchi batterie per Electric Vehicle e Hybrid Electric Vehicle di raggiungere nuovi livelli di precisione nella misurazione di un flusso di corrente in entrata ed in uscita dalla batteria. I produttori di auto necessitano di questo dato per massimizzare la percorrenza dei veicoli, per tenere le batterie in buone condizioni, e consentire l'uso di batterie delle dimensioni ottimali. Il trasduttore CAB di LEM offre una capacità di misurazione di corrente fino a ±400 A, con capacità di sovraccarico illimitato, e con una eccellente precisione: l'errore di Coulomb-counting durante il ciclo di guida è stato ridotto sotto lo 0,1%. È isolato galvanicamente e non è invasivo; nessuna necessità di connessione elettrica al circuito di alimentazione. Questo garantisce sicurezza di funzionamento e, contrariamente alle tecniche che utilizzano resistenze di rilevamento sull'alimentazione, non genera riscaldamento o perdite.

***CODICE MIP 2826988***

## Melexis amplia la sua gamma di sensori di posizione intelligenti su circuito integrato

***Melexis ha presentato il più recente prodotto facente parte della gamma di sensori di posizione su circuito integrato basati sulla sua tecnologia proprietaria Triaxis. Il dispositivo MLX90365 è un sensore completo con risoluzione di 12 bit che può essere usato in sistemi di rilevamento della posizione lineare o angolare. È caratterizzato da una funzione di trasferimento completamente programmabile con uscita selezionabile tra modalità analogica o con modulazione a larghezza d'impulso (PWM), funzioni di diagnostica integrate e capacità di rilevare condizioni di circuito aperto e corto circuito. Il sensore MLX90365 offre le stesse funzionalità di rilevamento del sensore MLX90364 presentato recentemente, che è realizzato nel particolare contenitore a due componenti stampati (Dual Mold Package). Il modello MLX90365 è realizzato invece nei classici contenitori per montaggio superficiale di tipo SOIC e TSSOP, che richiedono l'inserimento dei componenti passivi di protezione su una scheda a circuito stampato, mentre il modello gemello MLX90364 "senza PCB" può essere montato direttamente senza richiedere alcun circuito stampato supplementare. Il sensore su circuito integrato comprende anche il sistema di rilevazione automatica delle sovratensioni e sottotensioni.***

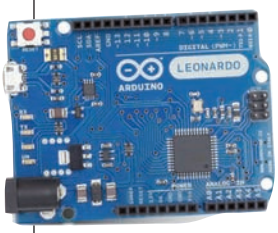



***CODICE MIP 2826990***

CODICE MIP 2815559

## L'OFFERTA MOUSER SI ARRICCHISCE DELLE SCHEDE DI SVILUPPO AVR LEONARDO DI ARDUINO

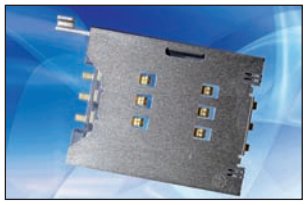
***Mouser Electronics, Inc ha annunciato la disponibilità a magazzino della tanto attesa scheda di sviluppo Leonardo AVR di Arduino. La scheda di sviluppo Arduino Leonardo AVR è un microcontroller basato su Atmel ATmega32u4. Include 20 I/O digitali, un oscillatore al cristallo da 16 MHz, una connessione micro USB, un jack di alimentazione, una testina ICSP e un pulsante di reset. La scheda Leonardo è diversa dalle precedenti in quanto l'ATmega32u4 offre comunicazione USB integrata, eliminando la necessità di un secondo processore. Leonardo viene quindi visto da un computer collegato come se fosse un HID, ad esempio mouse o tastiera, oltre che come porta seriale/COM (CDC). La scheda Leonardo contiene tutto il necessario per supportare il microcontroller. È sufficiente collegarla a un computer con cavo USB oppure alimentarla con un adattatore AC-DC o una batteria.***




***CODICE MIP 2826992***

## Lettore schede SIM ultrapiatto di nuova generazione

Yamaichi Electronics presenta con FMS006-8010-0, un lettore di schede con una nuova tecnologia di espulsione.
Con FMS006-8010-0 Yamaichi aggiunge una nuova versione alla sua gamma di lettori di schede di alta qualità, infatti è una variante del tipo "reversed mount" con un'altezza ultrapiatta di soli 1,34 mm. Un'altezza con un ingombro così ridotto rendeva finora possibile solo l'inserimento/l'estrazione manuale della scheda o con un procedimento indiretto d'inserimento/estrazione, per esempio tramite tray. Con lo sviluppo del nuovo meccanismo di espulsione è ora possibile estrarre anche schede ultrapiatte con maggiore comodità e sicurezza. Il meccanismo si attiva tirando una staffa, tramite la quale la scheda SIM viene spinta fuori dallo slot fino ad una certa altezza, garantendone il comodo prelievo. Le dimensioni esterne sono 17,5x21 mm. Il lettore di schede dispone di 6 contatti dorati. Come quasi tutti i lettori di schede Yamaichi, anche questo prodotto è dotato di un metodo di contatto a 2 punti (contatto a lamelle), concepito in modo tale da garantire la migliore sicurezza di contatto anche in caso di vibrazioni e scosse. Il lettore offre inoltre un'elevata sicurezza contro l'inserimento errato della scheda: una caratteristica di progettazione speciale la blocca automaticamente.



***CODICE MIP 2826740***

plug in

# CI PIACE VEDERTI SORRIDERE

md
millennium dataware
produciamo circuiti stampati

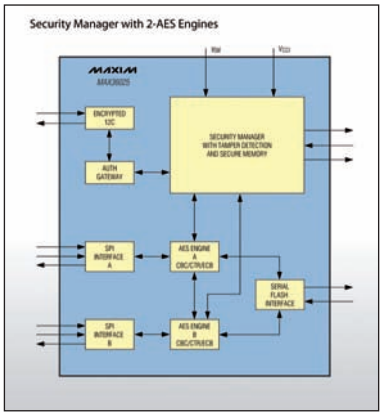
Da oggi realizziamo
**circuiti a 4 e 6 strati**
con l'aiuto
della tecnologia **OIR***
per un perfetto
allineamento
degli strati di rame,
il miglior laminato,
prodotto da Panasonic
e la garanzia **24 ore**
**o i circuiti sono gratis**
potete permettervi di lavorare
senza pensieri perché
**alla qualità ci pensiamo noi!**
**E I PREZZI,**
**SONO QUELLI DI SEMPRE!**

Inoltre sempre a vostra disposizione,
circuiti stampati a 1 e 2 facce,
su supporto di alluminio e lamine smd.

PER CHIARIMENTI, DETTAGLI
SULLE NOTE TECNICHE, ORDINI
**www.mdsrl.it**
PREVENTIVO ANONIMO, GRATUITO, IMMEDIATO

*(Optical Inner Layer Registration)

CODICE MIP 2815058




# Il Security Manager di Maxim garantisce la sicurezza delle smart grid

Maxim Integrated Products, Inc. ha annunciato l'introduzione di MAX36025, un controllore di nodo crittografico in grado di reagire alle manomissioni che integra due engine AES (Advanced Encryption Standard) hardware per garantire una cifratura più semplice e sicura. MAX36025 è un security manager (ovvero un integrato preposto alla gestione della sicurezza) che può essere collocato nei punti di accesso e/o di concentrazione dei dati di una smart grid per garantire la sicurezza in questi punti particolarmente critici di aggregazione dei dati. I due engine AES di MAX36025 trasferiscono le funzioni di cifratura a livello hardware, contribuendo in tal modo a ridurre drasticamente i tempi di progettazione e ad aumentare i livelli di sicurezza. I progettisti di sistemi quindi non si devono più preoccupare di scrivere codice di cifrature, processo questo che richiede tempi molto lunghi sia per la scrittura stessa, sia per le successive operazioni di debug e implementazione. Il software cifrato, inoltre, è vulnerabile in quanto eventuali errori (bug) non rilevati possono costituire la "porta d'ingresso" per un attacco da parte di virus. I due engine AES di MAX36025 possono essere utilizzati su canali indipendenti per proteggere differenti reti oppure utilizzati simultaneamente sul medesimo canale per effettuare una cifratura doppia.

***CODICE MIP 2826980***

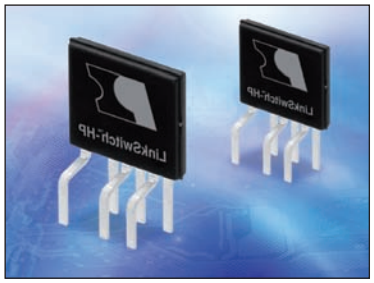
## Regolatori Buck ZVS Cool-Power Picor

Vicor introduce i nuovi regolatori buck ZVS Cool-Power Picor della serie PI33XX. I nuovi prodotti offrono ai progettisti di schede i massimi livelli di densità di potenza e di flessibilità, assicurando la possibilità di sviluppare soluzioni di regolazione point of load DC-DC ad alta efficienza. La presenza di un circuito ad alte prestazioni con architettura Zero-Voltage Switching (ZVS) all'interno dei prodotti della serie PI33xx permette di incrementare le prestazioni al punto di carico garantendo livelli di efficienza da record, che possono raggiungere picchi del 98%. I prodotti della serie PI33xx sono altamente integrati: all'interno di un SiP LGA ad alta densità da 10 x 14 x 2,56 mm includono circuito di controllo, semiconduttori di potenza e componenti di supporto. I regolatori buck PI33xx sono in grado di convertire tensioni di ingresso da 8 V a 36 V su valori di uscita da 1 V e 16 V, con correnti fino a 10 A, erogando una potenza che arriva a toccare i 120 W. La potenza erogata può essere ulteriormente aumentata collegando fino a sei regolatori buck PI33xx con uno schema di condivisione di corrente single wire, senza ricorrere ad ulteriori componenti.

***CODICE MIP 2826984***

## Nuovi circuiti integrati LinkSwitch-HP di Power Integrations

**Power Integrations ha annunciato LinkSwitch-HP, una nuova famiglia di circuiti integrati con switch offline a basso consumo che fornisce fino a 90 W di potenza in uscita con regolazione precisa sul lato primario (PSR).**



**Per determinare la quantità di potenza che il fronte primario deve fornire al secondario isolato, i circuiti integrati di LinkSwitch-HP utilizzano algoritmi di controllo innovativi, le proprietà del trasformatore di potenza principale e il diodo in uscita, anziché gli optoaccoppiatori e i circuiti di feedback associati. Questo metodo riduce notevolmente il numero dei componenti, l'ingombro e i costi, aumentando al contempo l'affidabilità.**

**Divenuta popolare più di dieci anni fa con la release della linea LinkSwitch di Power Integrations, la regolazione PSR è attualmente lo standard di molte applicazioni a bassa potenza come i caricabatterie dei telefoni cellulari; finora, però, questa tecnologia non offriva la precisione di tensione costante necessaria ai prodotti con potenza più elevata. I dispositivi LinkSwitch-HP sono dotati di un'architettura di controllo multi-mode esclusiva che segna un progresso decisivo nell'evoluzione della regolazione PSR, rendendola un approccio ideale per le applicazioni di alimentazione con potenza tra 9 W e 90 W.**

***CODICE MIP 2826986***

# Expo Elettronica

MOSTRA MERCATO DI

- > ELETTRONICA
- > INFORMATICA
- > TELEFONIA
- > DVD > GAMES
- > HOBBISTICA

PREZZI SEMPRE PIÙ BASSI!

Calendario Ottobre/Dicembre 2012

| | |
|---|---|
| FAENZA (RA) | 20-21 OTTOBRE |
| BASTIA UMBRA (PG) | 27-28 OTTOBRE |
| MORCIANO DI ROMAGNA (RN) | 3-4 NOVEMBRE |
| FORLI' (FC) | 1-2 DICEMBRE |

CODICE MIP 2815141

rif. Fare Elettronica

INFO, NEWSLETTER E TICKET RIDOTTO WWW.EXPOELETTRONICA.IT

DIVENTIAMO AMICI SU WWW.FACEBOOK.COM/EXPO.ELETTRONICA

con il patrocinio di

Ministero dello Sviluppo Economico

organizzato da:

BLU Nautilus tel 0541 439573 www.blunautilus.it

Info e ticket ridotto:

www.expoelettronica.it

## EXPO RADIO POTENZA

*L'Expo Radio & Informatica, la mostra mercato dedicata all'elettronica e all'informatica, compagni ormai indispensabile nel lavoro e negli hobby. Ideata per consentire agli operatori del settore una migliore valutazione dell'offerta delle nuove tecnologie. Inoltre offre anche ad un pubblico più eterogeneo, in cerca di buone occasioni commerciali tutta l'esperienza e la professionalità acquisita nel campo delle telecomunicazioni e dell'informatica. In contemporanea è prevista una Mostra di Protezione Civile, per sottolineare l'importanza delle comunicazioni radio in eventi calamitosi, indispensabili nella gestione dell'emergenza. Saranno presenti espositori provenienti dall'intero territorio nazionale con: Radioricetrasmettitori per CB – OM, Computer, Telefonia, Antenne e Parabole per TV Sat, Hi-Fi Car, Strumenti di misura e utensili vari, Surplus, Radio d'epoca, Editoria Specializzata.*

*Dove: Potenza, area industriale di Tito Scalo*
*Quando: 12-14 Ottobre 2012*
*Orari: 9.00/13.00-16.30/20.30*
*Organizzazione: ENTE FIERA AUTONOMO BASILICATA*
*Info: www.basilicatafiere.it*

**CODICE MIP 2810469**

## RADIANT AND SILICON

Sin dalle primissime edizioni ha goduto del patrocinio della sezione A.R.I. (Associazione Radioamatori Italiani) di Milano, del supporto dell'A.I.R.E. (Associazione Italiana per la Radio d'Epoca), dell'A.I.R. (Associazione Italiana Radioascolto) e della prestigiosa partecipazione di importanti Enti Pubblici come la Provincia di Milano, presente con le sue postazioni mobili del Servizio di Protezione Civile e, ultimamente, il Ministero delle Comunicazioni. La Rassegna si configura in tre settori: mostra-mercato (apparati e componenti per telecomunicazioni, Internet e ricetrasmissioni di terra e satellitari, antenne, elettronica, informatica, telefonia statica e cellulare, hobbistica ed editoria); borsa-scambio (surplus radioamatoriale, informatico ed elettronico); radioantiquariato (radio d'epoca, grammofoni, valvole, ricambi, dischi in vinile, schemari ed editoria specializzata). Attualmente RADIANT, che si sviluppa su una superficie espositiva superiore ai 10.000 mq. coperti, con più di 180 espositori per edizione ed oltre 40.000 visitatori annui, è considerata la più importante Mostra-Mercato nazionale del suo settore. Fra le categorie merceologiche ricordiamo: elettronica, informatica, telefonia, radiantismo, editoria, tv satellitare, hobbistica, surplus, radio d'epoca.

Dove: Parco Esposizioni Novegro, Segrate (MI)
Quando: 22-23 Settembre 2012
Orari: Sabato 9.00/18.00 - Domenica 9.00/17.00
Organizzazione: COMIS
info: www.parcoesposizioninovegro.it

**CODICE MIP 2810497**

## Makers Italy 2012

Grande evento italiano dedicato al mondo dei makers, dei creativi digitali e tecnologici, dei "garage innovators", ma anche a tutti gli appassionati del Diy (do-it-yourself) che, con le loro innovazioni e i loro nuovi modelli di business, stanno dando vita a quella che molti commentatori non esitano a definire una nuova "rivoluzione industriale". Una rivoluzione che vede protagonista la cultura digitale, la quale, dopo aver sovvertito il mondo dei bit e quindi l'editoria, la musica e i video attraverso Internet, secondo Chris Anderson, direttore di Wired Usa, «ora sta per trasformare il mondo degli atomi, quindi degli oggetti fisici», offrendo infinite opportunità imprenditoriali a giovani creativi e brillanti, che saranno impegnati a generare business partendo dall'open source e dalla collaborazione tra persone.

Dove: Fiera Milano Rho (MI)
Quando: 9-11 novembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: Italia Crea
info: www.makersitaly.it

**CODICE MIP 2826019**

## Automazione in fiera

SAVE è un appuntamento innovativo che unisce una parte espositiva in fiera ad una forte componente formativa. Area espositiva dove incontrare agli stand i principali leader di settore, centinaia di convegni e workshop accessibili per gli operatori qualificati, cinque eventi internazionali e un evento speciale in contemporanea, gli operatori professionali accedono gratuitamente alla mostra e usufruiscono di tutti i servizi. L'esigenza percepita oggi è rendere adeguate le modalità fieristiche, renderle attuali e sempre più proficue per creare una fiera sull'automazione e sulla strumentazione utile agli operatori.
Dove: Verona
Quando: 24-25 ottobre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: EIOM
info: www.exposave.com

***CODICE MIP 2823210***

## ELECTRONIC DAYS

***Electronic Days, Fiera dell'Elettronica in genere, Mostra Mercato con vendita consentita.***
***Dove: Mantova, presso PalaBam, Via M.Gioia Zona Favorita***
***Quando: 17-28 Ottobre 2012***
***Organizzazione: ITALFIERE srl***
***info: www.italfiere.net***

***CODICE MIP 2810473***

## Fiera dell'informatica elettronica e radiantismo

***Grandi occasioni per tutti, a partire dai supporti per registrazioni audio/video per finire con l'informatica, in continua crescita e il radiantismo. Uno spazio considerevole, con più di trenta espositori, è stato riservato all'hobbistica e all'elettronica con la mostra scambio del radioamatore. Anche gli strumenti di misura e le apparecchiature elettroniche sono ben rappresentate dalla presenza di molti espositori specializzati.***

***Dove: Rovigo***
***Quando: 27-28 Ottobre 2012***
***Orari: Sabato 9:00/19:00***
***Organizzazione: AREA REBUS***
***info: www.arearebus.com/fiera***

***CODICE MIP 2810495***

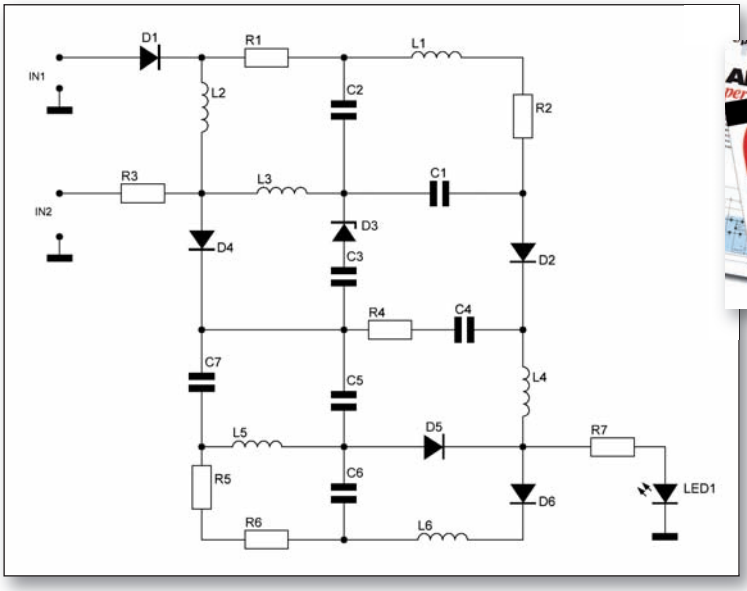

FIERE ELETTRONICA
INFORMATICA E RADIANTISMO
con Mercatino Radio Amatoriale e Surplus
PIACENZA
PIACENZA EXPO'
LOCALITA' LE MOSE
15-16 SETTEMBRE
ORARIO CONTINUATO 9.00 - 18.30
RAVENNA
PALA DE ANDRÈ
Viale Europa 1
29-30 SETTEMBRE
ORARIO CONTINUATO 9.00 - 18.30
PARCHEGGIO GRATUITO
PROSSIMI APPUNTAMENTI
13-14 OTTOBRE
FIERA DI FERRARA
VIA DELLA FIERA 11
3-4 NOVEMBRE
BOLOGNA
PALA NORD - PARCONORD
15-16 DICEMBRE
(DA CONFERMARE)
FIERA DI PIACENZA
PIACENZA EXPO' - LOCALITA' LE MOSE
Cell. 348 9030832 - www.mondoelettronica.net
Cosmohelp
VALE COME RIDOTTO ANCHE IN FOTOCOPIA
FARE ELETTRONICA
CODICE MIP 2815238

# elettro quiz 327

# rispondi & vinci

*Le risposte ai quiz "facile" e "difficile" vanno inviate esclusivamente compilando il modulo su www.farelettronica.com/eq specificando la parola chiave* ***"Coil"****.*

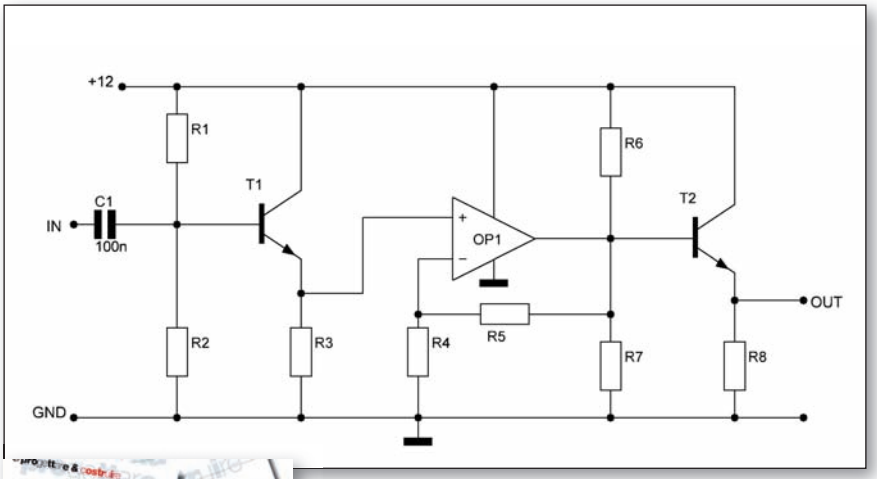
*Le risposte e i vincitori (previa autorizzazione) sono pubblicati alla pagina www.farelettronica.com/eq a partire dal 15 del mese successivo alla pubblicazione sulla rivista. A tutti i partecipanti verrà assegnato un buono sconto del 10% (validità 3 mesi dalla data di assegnazione) utilizzabile per un prossimo acquisto su www.ieshop.it*



**abbonamento al club di Fare Elettronica**

## facile

*A quale dei due ingressi andrà applicata una tensione di 20 V affinché si accenda il LED in uscita?*



**Abbonamento a Fare Elettronica**

## difficile

*Con riferimento al circuito di figura, quali delle seguenti affermazioni sono vere?*

***a)*** *Il circuito svolge la funzione di amplificatore;*

***b)*** *La tensione di uscita ha una componente continua;*

***c)*** *L'uscita è ad alta impedenza;*

***d)*** *Il guadagno del sistema dipende pesantemente dalle resistenze R4 e R5;*

***e)*** *Il segnale di uscita è invertito rispetto a quello di ingresso.*

# Fiera dell'Elettronica e del Radioamatore

Fiera Nazionale.
Hardware, software, componentistica, telefonia ed elettronica di consumo

**29 e 30 settembre 2012**
dalle ore 8.30 alle 18
Padiglioni Fiera Millenaria
Gonzaga (Mantova)

fachiro.com

FIERA 1000 NARIA

**Fiera Millenaria di Gonzaga** srl
Via Fiera Millenaria 13, Gonzaga (MN)
T +39 0376 58098 F +39 0376 528153
info@fieramillenaria.it **www.fieramillenaria.it**

CODICE MIP 2817573

DI PIERO BOCCADORO

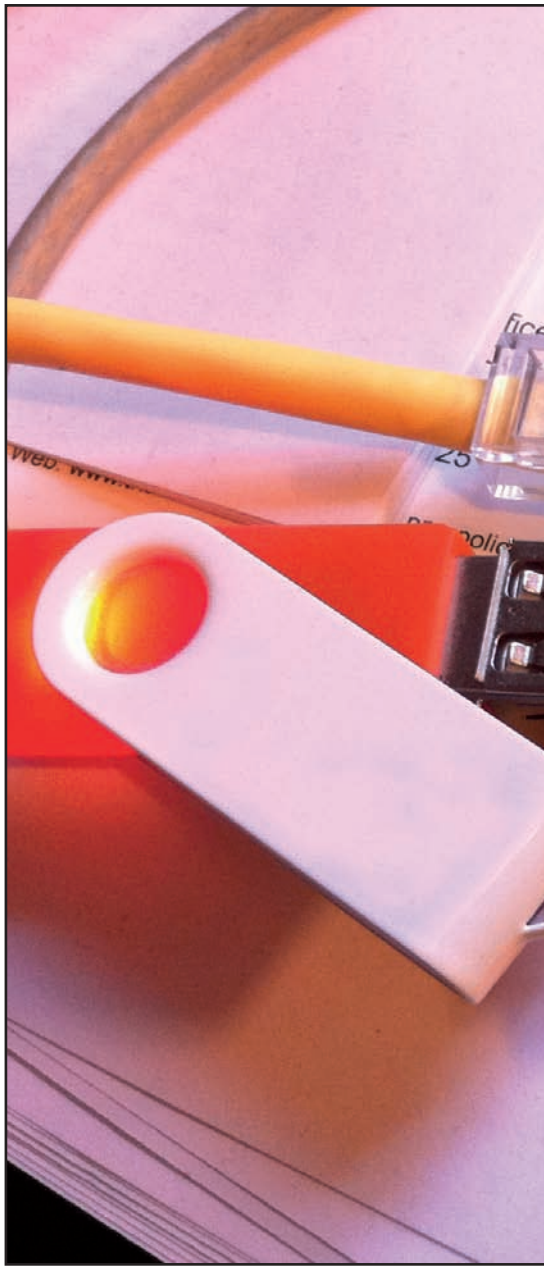
# PC EMBEDDED: *il progetto* RASPBERRY PI

*(parte prima)*

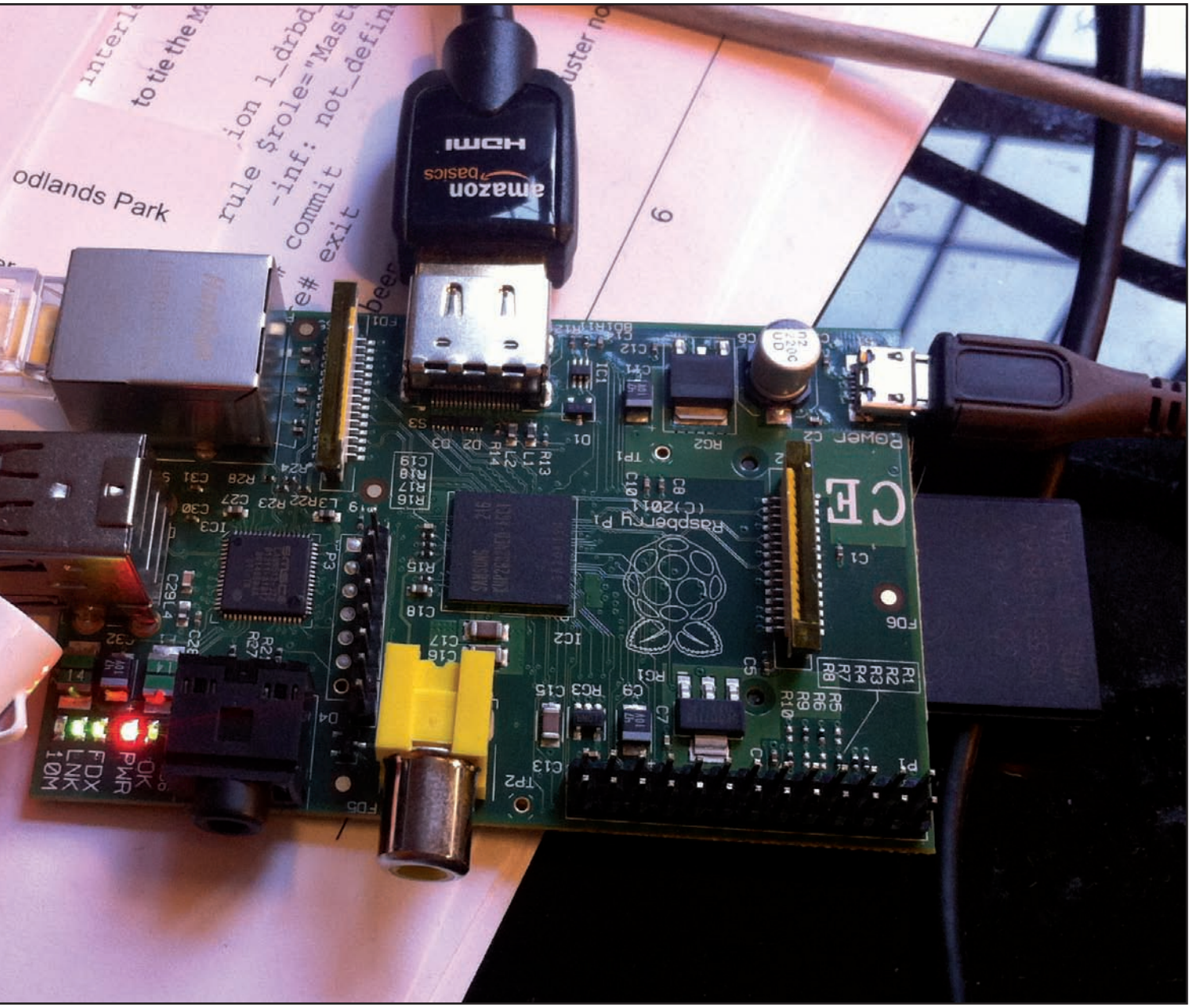
**In un mondo in continuo cambiamento, anche il personal computer si evolve e diventa pocket, con il nuovo Raspberry Pi. In questo articolo introduttivo vediamo innanzitutto come il Raspberry Pi si pone nel mercato dei PC embedded**

Un sistema di tipo embedded rappresenta il risultato del processo di integrazione e di implementazione di funzioni comunque complesse all'interno di sistemi di elaborazione di dati a microprocessore. Tipicamente, questo tipo di sistemi è pensato per applicazioni specifiche e questo lo rende diverso anche da altri progetti che rientrino in questa categoria. Lo spirito con cui nasce un sistema embedded è quello di provare a dare una risposta altamente specificata, ovvero orientata a un'applicazione in particolare, potendo contare sul fatto che al sistema non verrà mai chiesto di fare altro. Così, insomma, quella che si progetta è una piattaforma hardware ideata ad hoc e che assume conformazioni minimaliste ma certamente in grado di svolgere il compito specificato.
Esistono diverse applicazioni e svariate tipologie di processi che possono beneficiare di questo tipo di "mentalità" e queste si differenziano sulla base del fatto che i sistemi assumeranno dimensioni e tipologie differenti. Questo è conseguenza del fatto che il sistema verrà dimensionato in relazione al particolare tipo di microprocessore implementato; da ciò ne discende che il sistema operativo, la complessità del software e il numero e il tipo delle connessioni o dei protocolli di comunicazione di cui dovrà essere dotato saranno molto diversi.
Non solo la progettazione risulta altamente specifica ma anche il test di funzionamento e le verifiche saranno coerentemente pensate per dimostrare che solo quella particolare applicazione funzioni in maniera ottimale.
Potendo ridurre il numero di componenti presenti all'interno del sistema al mini-

mo indispensabile per lo svolgimento dello specifico compito assegnato, una delle variabili che ne beneficia maggiormente è il tempo di elaborazione il quale tende ad essere sempre più breve dando "l'illusione" di lavorare in real-time.
Dicevamo, dunque, che esistono diversi tipi di applicazioni: il primo sistema ha addirittura permesso di viaggiare nello spazio, altri, più avanti, hanno permesso la realizzazione di guide inerziali. La diffusione dei circuiti integrati, sia per tipo sia per aumento della precisione, e l'abbattimento dei costi dei processi tecnologici hanno permesso di rendere questo approccio progettuale applicabile in diversi campi della tecnica.
Così, dagli anni Sessanta a oggi, i sistemi embedded si sono evoluti partendo da piccole soluzioni implementate in sistemi più ampi fino a rappresentare essi stessi il sistema. E oggi, la maggior parte dei sistemi di questo tipo viene progettata quando è necessario che una stessa "azione", uno stesso procedimento, si siano ripetuti più volte. Esempi di strutture sono i bancomat, i distributori di bevande, gli erogatori nelle stazioni di servizio, stampanti e fotocopiatrici ma anche le centraline delle automobili (che gestiscono i vari sistemi automatizzati come ESP, ABS, ESC), o alcuni apparecchi biomedicali come gli ecografi. In pratica, una diffusione capillare di sistemi che possono, all'occorrenza, anche essere ri-programmati qualora necessario.
In linea del tutto generale, risulta difficile caratterizzare la velocità o i costi di un sistema embedded; tuttavia, soprattutto quando trattiamo di sistemi che devono processare una grande quantità di dati, il progetto diventa la parte più "esosa" dell'intero processo. Nella stragrande maggioranza dei sistemi embedded le specifiche possono essere soddisfatte grazie a una combinazione di hardware dedicato a specifiche funzioni e poche righe di codice ottimizzate per svolgere i singoli compiti. Questa affermazione spiega per quale motivo una semplice scheda dalle dimensioni contenute sia in grado di garantire funzionamento e autonomia a macchine che possono essere lasciate a operare sul campo anche senza l'intervento dell'operatore per una periodica revisione, salvo che non sia previsto. Ad



esempio, salvo che non vengano maltrattate, come spesso accade, gli apparecchi dispenser di bibite o snack non necessitano della presenza dell'operatore se non per il rifornimento della merce e lo scarico della cassa in gettoni.
Un esempio di questo tipo di applicazioni è il decoder per la televisione satellitare che, nonostante sia un sistema pensato per lavorare su una ingente mole di dati, possiede hardware dedicato che identifica le sequenze ordinate, decodifica il flusso di dati e indirizza il tutto a video.
L'architettura, vista nel suo complesso, risulta, in definitiva, più semplice rispetto a




quella di un PC "tradizionale", anche a parità di operazioni da svolgere. Esse possono essere effettuate dalla macchina grazie alla presenza di un sistema operativo snello e progettato, concordemente con l'hardware, in maniera mirata e che può operare, come si diceva in precedenza, in realt-time. Non è escluso che, se il sistema ha necessità differenti o comunque complesse, si possa rendere necessario l'utilizzo di una CPU a elevate prestazioni oppure l'utilizzo di memoria flash, come in applicazioni consumer quali lettori portatili o fotocamere.
Il software di cui stiamo parlando, snello come accennato in precedenza, è un firmware scritto su una ROM oppure su una EEPROM. La scelta del tipo di memoria da utilizzare dipende, anch'essa, dalla particolare applicazione. Infatti, se abbiamo a che fare con sistemi che non sono pensati per soddisfare all'esigenza di una possibile ri-programmazione, come un UPS, non è necessario che la memoria su cui scriviamo permetta la cancellazione e la riscrittura del firmware stesso. Tuttavia, è evidente, questa possibilità rende il sistema più versatile e, in definitiva, più appetibile.
Lo sviluppo di questi sistemi può comportare procedure e tempi tra i più disparati, in dipendenza, è chiaro, dell'applicazione cui sono mirati. Esistono sistemi embedded che vanno inseriti all'interno di processi industriali e sistemi che saranno deputati al controllo degli stessi. In questo caso, comanderanno o programmeranno azioni correttive o attiveranno procedure conseguenti. La programmazione di questi sistemi, si intuisce, necessiterà di tool di sviluppo di complessità variabile e crescente con il grado di criticità delle condizioni operative del sistema finito. È possibile che lo sviluppo avvenga con PLC o micro-

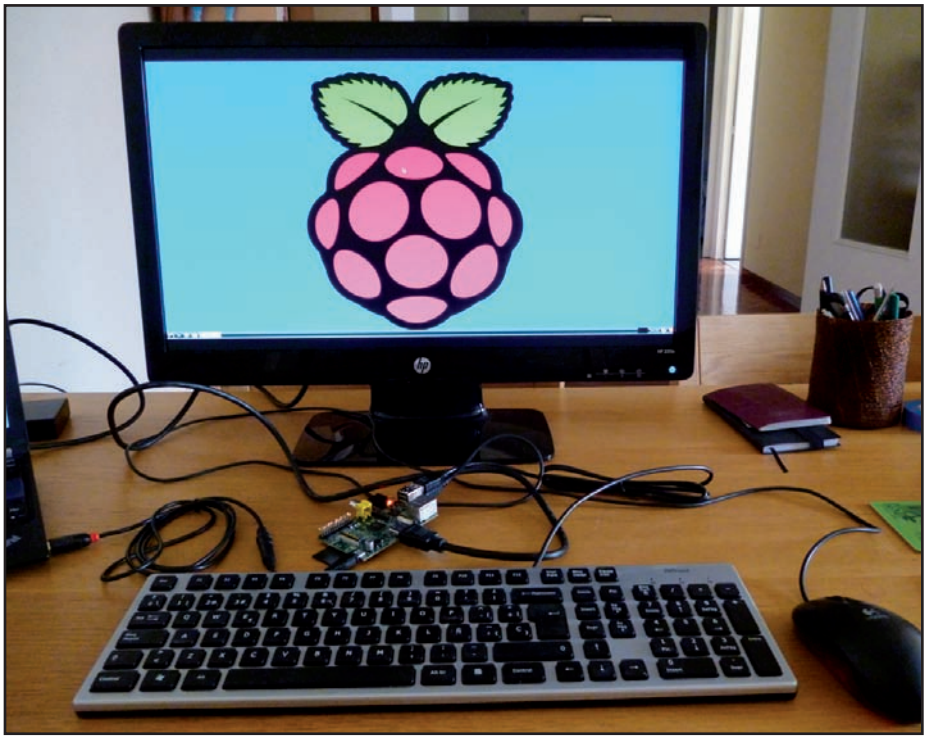
controllori e vengono utilizzate piattaforme di sviluppo basate, tra le altre, su architetture ARM, IBM compatibili, X86, Atmel. Col crescere della complessità del sistema si rende, inoltre, indispensabile l'interfacciamento con ADC e DAC, oscillatori, DSP e dispositivi di memoria dedicati; questo suggerisce uno sviluppo orientato verso le soluzioni SoC (acronimo che indica i System on a Chip). Un esempio della loro funzionalità è rappresentato da un esponente che sta facendo molto parlare di sé: l'Apple A5. Accompagnato dalla sua variante A5X, si tratta di un processore di tipo PoP (package on a package) SoC che è stato progettato da Apple, prodotto da Samsung e implementato nell'iPad 2, oltre che nell'iPhone 4S, con i risultati che sono sotto gli occhi di tutti. Il sistema è piuttosto stabile, affidabile ed è in grado di gestire in apparente scioltezza anche animazioni ed effetti complessi ormai diventati classici.

Il culmine della complessità di questi sistemi si raggiunge, comunque, quando per effettuare lo sviluppo risulta indispensabile utilizzare FPGA (Field-Programmable Gate Array) che, a dire il vero, vengono progettati proprio allo scopo di effettuare lo sviluppo di sistemi come questi direttamente sul campo, anche grazie all'uso di circuiti di interfacciamento specifici.

Come accade per il software, i progettisti di sistemi embedded utilizzano compilatori e linguaggi di programmazione che dipendono dalla specifica piattaforma. Anche i debugger hanno necessità di essere altamente specializzati e scelti in dipendenza dalla piattaforma sulla quale si decide di programmare il firmware stesso. Tra le tante soluzioni possibili ne spiccano, certamente, alcune, tra cui c'è ICE (In-Circuit Emulator), un dispositivo hardware che sostituisce, o si interfaccia con, il microprocessore, e offre funzionalità adatte a caricare velocemente il codice ed effettuare il debug dello stesso all'interno del sistema.

Esistono anche soluzioni rivolte a sistemi che implementano i DSP (Digital Signal Processing), ovvero MathCAD o Mathematica.

La varietà infinita di questi sistemi chiarisce perché in un solo, breve articolo, sarebbe impossibile esaurire l'intera gamma disponibile. In generale, però, possiamo dire che, a seconda del campo applicativo, è di norma, possibile trovare diverse soluzioni tra compilatori e linker specifici. Un sistema embedded può avere, inoltre, un suo linguaggio specifico, esattamente in analogia con quanto accade per Ar-



fare elettronica

# Vieni a trovarci su Facebook! unisciti alle migliaia di fans!!!

aggiornamenti e novità in tempo reale

articoli gratuiti per i fan

anteprime e anticipazioni

eventi

contatti con gli autori

inviaci i tuoi consigli e i tuoi commenti

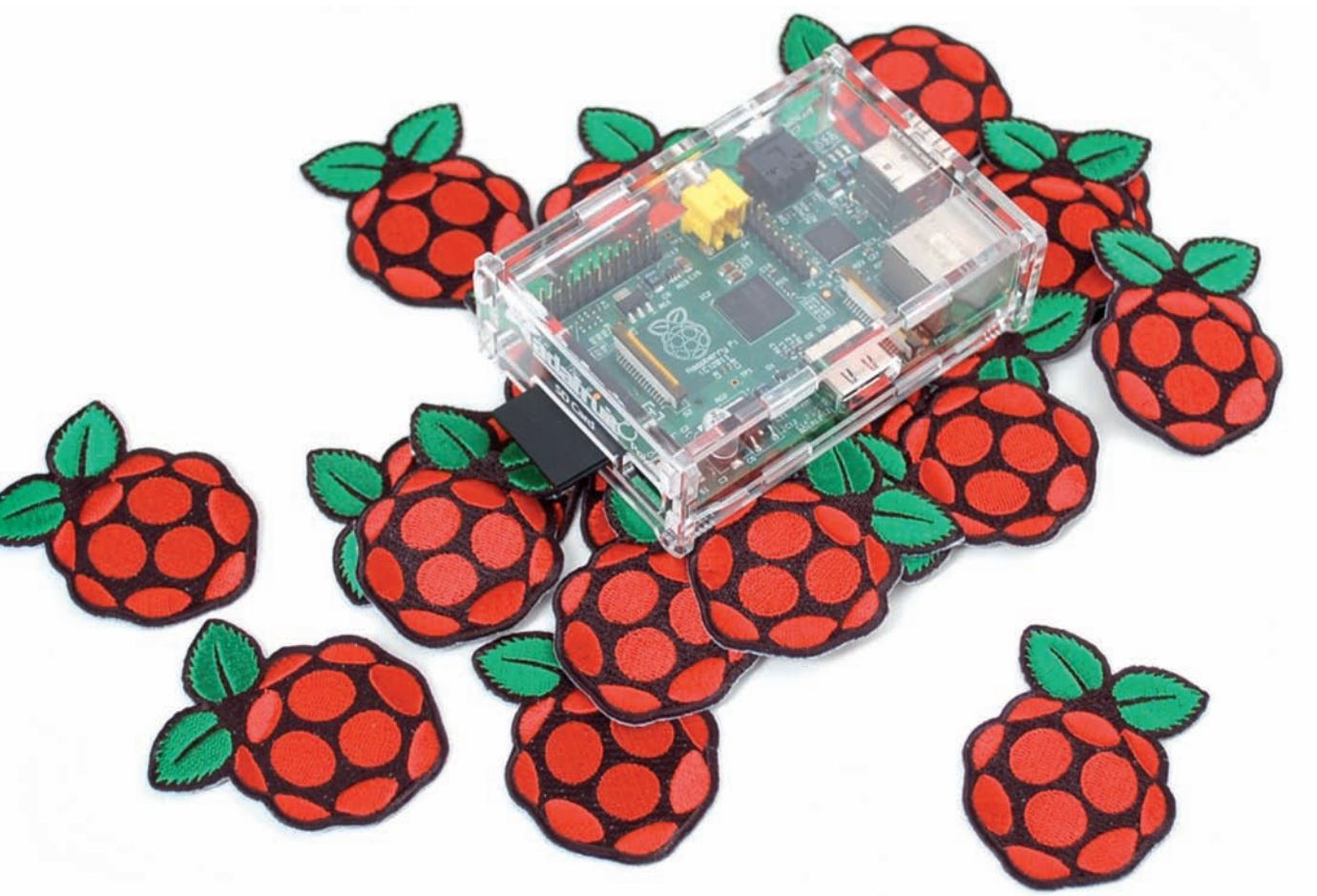
condividi i contenuti con i tuoi amici

**DIVENTA PROTAGONISTA DELLA TUA RIVISTA PREFERITA!**

*www.facebook.com/farelettronica*

CODICE MIP 2827171

duino, Processing e Wiring. Il compilatore, in questo caso, si rende interprete del linguaggio pseudo-C utilizzato nella programmazione per declinare le specifiche istruzioni al microcontrollore ed è proprio questa operazione che assume connotazioni diverse a seconda del caso applicativo.
Questo esempio, così come tutti gli altri analoghi che si potrebbero fare, dimostra come i programmi per la generazione del codice possono avere provenienza diversa. Una differenziazione, o meglio una precisazione, che va fatta immediatamente è che esistono tool provenienti dal progetto GNU ed esistono software e linguaggi completamente proprietari (e i due esempi fatti finora sono stati scelti ad hoc perché esponenti notevoli delle relative categorie). Scegliere un linguaggio, un tool e un debugger cross-platform non chiuso rappresenta un'opportunità di sviluppo e di interfacciabilità che oggi non è più una scelta coraggiosa ma intelligente. Se solo si pensa che sistemi come Android hanno fatto della disponibilità una bandiera ci si rende facilmente conto che la scelta si è dimostrata ampiamente vincente e di come perfino il mercato libero ripaghi tale politica.
Su microcontrollore, inoltre, quando le soluzioni sono "semplici", viene, tipicamente, eseguito ciclicamente un unico programma di pochi byte. Man mano che il sistema diventa più complesso, però, è possibile che si richieda l'impiego degli stessi sistemi operativi che vediamo ogni giorno su PC tradizionali, magari in versione "lite" per ottimizzarne la funzionalità garantendone le feature "indispensabili".
Prima di passare all'argomento di oggi, il Raspberry Pi, è importante fare una precisazione sulle politiche di gestione dei sistemi operativi implementati su queste macchine. In particolare, questi sistemi hanno politiche e capacità di auto-verifica che fanno parte delle proprie funzionalità. Se in un sistema operativo "classico" esiste la possibilità di abilitare funzioni di reboot forzato o di "sblocco" qualora un'applicazione o più vadano in crash, questi sistemi implementano politiche di protezione dette "watchdog". Questo vero e proprio cane da guardia monitora l'operato della scheda e termina i processi che non rispondono per lungo tempo, che durano più del previsto e/o che assumono "comportamenti" comunque anomali. Anche questo tipo di auto-regolamentazione deve essere declinata nel particolare sistema in analisi.
Tra le operazioni svolte in tal senso generalmente ci sono: verifica del calcolatore (con controllo per CPU, RAM ed eventuali altri dispositivi di memoria programmabile), verifica delle periferiche (se presenti), verifica dell'alimentazione (tipicamente anche un calo di tensione ha necessità di essere gestito, specie in condizioni operative estreme). E inoltre verifiche di cablaggi, di comunicazione, di interfaccia, di consumi, di operatività ed eventualmente di sicurezza.
In questo complesso e variegato scenario nasce il progetto di cui parliamo oggi, il Raspberry-Pi. Si tratta di un computer embedded dalle dimensioni piuttosto contenute (circa quelle di una carta di credito) che può essere collegato molto facilmente a una serie di periferiche quali TV, mouse, tastiera e altre. Fatto questo, la scheda diventa in grado di svolgere egregiamente le funzioni di un normale PC desktop utilizzato per applicativi da ufficio: navigazione su Internet, stesura di documenti e addirittura videogiochi (assolutamente non di ultima generazione!) che possono essere ospitati dalla scheda. La navigazione su Internet diventa un'applicazione da salotto con la possibilità di vedere video in alta definizione e garantire l'accessibilità anche ai più piccoli.
Sulla scheda viene impiegato un System-on-a-chip Broadcom BCM2835 equipaggiato con un core ARM11, in particolare parliamo del 76JZFS, che gira a 700 MHz e dimostra, tra le caratteristiche più importanti un ridotto consumo di potenza. Tra le feature della scheda compare anche un core grafico che gestisce OpenGL per l'accelerazione grafica, che permette anche la visualizzazione di filmati a 1080p.
Esso si propone, dunque, come un sistema facile e immediato, funzionale e completo, disponibile anche in modelli differenti. Esistono, infatti, una versione A e una B, che si differenziano principalmente per l'ammontare di memoria RAM disponibile (256 MB sul modello A e 512 MB sul modello B). Le dimensioni contenute della scheda, il suo peso esiguo e la



grande funzionalità rendono questo sistema accattivante. Un po' meno accattivante il fatto che, se non siete esperti, probabilmente non gradirete molto il fatto che la scheda venga venduta "nuda". Tuttavia, se questo vi mette a disagio, è possibile acquistare separatamente opportuni case, anche in colori differenti.
A proposito di acquisti, parliamo della distribuzione; se leggere queste poche righe vi ha fatto venir voglia di procurarvene uno, sappiate che Raspberry Pi è in distribuzione presso Element 14 / Farnell e RS Online anche se sono previsti tempi di attesa diversificati per i vari mercati e le varie piazze, così come diversi sono i costi indicati per ciascuna regione. La scheda ha prezzi che si aggirano intorno a 35 euro ma questa è una cifra indicativa sul mercato europeo; il lancio era stato previsto per 25 dollari ma sono previsti prezzi differenti anche per il mercato anglosassone e altre nazioni.
Tuttavia, le oscillazioni di prezzo sono "contenute" al punto da non inficiare l'evidente vantaggio, o meglio lo scopo, della scheda: avere a disposizione una soluzione PC completa embedded che non occupi grande spazio ma che sia in grado di far lavorare tutti coloro che hanno bisogno di gestire applicazioni "di base" a costi relativamente irrisori. E ovvio, e forse è il caso di precisarlo, che nessuno si sognerebbe mai di utilizzare questa scheda per – diciamo – realizzare montaggi video di caratura professionale ma queste non sono, evidentemente, applicazioni "di base".
E passiamo alla vera chicca del Raspberry Pi: il sistema operativo, una caratteristica che farà sorridere di gioia tutti gli amanti del software libero. Il sistema supporta appieno, e così viene raccomandato direttamente dai produttori, Linux e in particolare Debian, un'incarnazione decisamente user-friendly del sistema ideato da Linus Torwalds. E poiché avere Linux senza Internet è come farsi lo shampoo senza capelli, una semplice chiavetta Wi-Fi connessa tramite usb servirà allo scopo.
Nulla di nuovo, insomma, direte voi: alla fine si tratta di un computer qualunque. Beh, non proprio. Perché questa scheda fa tutto in spazi contenuti minimizzando i problemi inerenti la dissipazione della temperatura e rappresenta una configurazione minimalista ma nel contempo versatile per esigenze, come detto, di base ma variegate.
Sarà anche possibile provare a realizzare dei prototipi e delle sperimentazioni, grazie ai pin che la scheda ha disponibili per l'utente.
E se avete ancora dubbi, fate un giro sul sito http://www.raspberrypi.org, dove troverete tutte le informazioni che vi serve conoscere per decidere se questa è una scheda che vi interessa oppure no.
E chissà che questo piccolo capolavoro della tecnica non si candidi a superare addirittura il successo di Arduino ed emuli.
Per scoprire tutte le caratteristiche e le potenzialità di questa piattaforma, nel prossimo numero vi proporremo un'analisi tecnica delle prestazioni della scheda, anche in raffronto ad altre concorrenti, e vi guideremo nella progettazione hardware e programmazione per giungere alla realizzazione di applicazioni veramente interessanti. Non perdete quindi i prossimi numeri di Fare Elettronica! ❑

***CODICE MIP 2827030***

# CAMPAGNA ABBONAMENTI

# PROMO tutte 3

**RISPARMIO ASSICURATO!**
Risparmierai fino a 73,00 EUR sul prezzo di copertina. Garanzia del prezzo bloccato per un anno. Le riviste ti saranno recapitate comodamente a casa. Riceverai un buono sconto del 20% per un tuo prossimo acquisto su www.ieshop.it

## UNA PROMOZIONE ESCLUSIVA:

- 11 numeri di FARE ELETTRONICA
- 11 numeri di ELEKTOR
- 11 numeri di FIRMWARE (edizione digitale) compreso l'accesso all'archivio a partire dal numero 49 di febbraio 2010
- 1 anno di accesso al CLUB di FARE ELETTRONICA
- 1 anno di accesso al CLUB di FIRMWARE,
- 20% di sconto per un acquisto su www.ieshop.it

**TUTTO A SOLI € 119,50** ANZICHE' € 192,50

## Sì, mi abbono alla PROMOTUTTE3 (è possibile abbonarsi online www.ieshop.it/promotutte3)

Riceverò 11 numeri di Fare Elettronica e 11 di Elektor in edizione cartacea, potrò accedere all'archivio di Firmware in edizione digitale a partire dal numero 49 di febbraio 2010 e ai prossimi 11 numeri, compreso un anno di accesso al CLUB di Fare elettronica e Firmware, tutto a **soli € 119,50** anziché €192,50

**METODI DI PAGAMENTO**

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . . .
NUMERO . . . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . . . .

**INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, via Giotto 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225**
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . Cognome . . . . . .
Indirizzo . . . . . n° . . . . . Cap . . . . . Città . . . . . Prov . . . . . .
Tel . . . . . Fax . . . . . Email . . . . . .
Ragione Sociale . . . . . Codice fiscale . . . . . Partita IVA . . . . . .

DATA . . . . . FIRMA . . . . . ☐ Fattura

1

## PERCHÉ SCEGLIERE FARE ELETTRONICA:

LAVORO: Se lavori come progettista elettronico, Fare Elettronica saprà esserti utilissima con numerose idee e spunti sia dal punto di vista pratico che teorico. DIDATTICA: se lavori come insegnante in una scuola ad indirizzo tecnico, Fare Elettronica risulterà tra le migliori fonti di ispirazione per te e per i tuoi allievi. AGGIORNAMENTO TECNOLOGICO: se sei alla ricerca di una fonte d'ispirazione, Fare Elettronica non ti deluderà, fornendoti l'aggiornamento sulle più recenti tecnologie, trattate in modo esauriente e competente. HOBBY: se invece l'elettronica è la tua passione, Fare Elettronica ti permetterà di coltivarla giorno per giorno con progetti sempre appassionanti ed alla tua portata.

2

## PERCHÉ SCEGLIERE FIRMWARE:

Firmware è una fantastica rivista digitale, tra le prime uscite nel settore dell'elettronica e la prima in assoluto in lingua italiana. Nasce come naturale evoluzione della rivista Firmware in edizione cartacea, rivista dedicata ai professionisti dell'elettronica, esigenti e competenti bisognosi di uno strumento di aggiornamento e di know how per l'elettronica professionale. Firmware è basata su una piattaforma che saprà regalarti una nuova esperienza di accesso a incredibili contenuti. Nessuna rivista ti sembrerà più all'altezza dopo aver visto Firmware!

3

## PERCHÉ SCEGLIERE ELEKTOR:

Elektor è in circolazione in oltre 50 Paesi con un totale di 300,000 copie mensili. Pubblicato in Inghilterra, Francia, Paesi Bassi, Spagna, Grecia, Portogallo, Brasile, Svezia, Finlandia, Germania, Italia, Stati Uniti. La rivista Elektor è distribuita a livello mondiale tramite abbonamento, edicole e librerie. I siti internet sono tradotti in Inglese, Tedesco, Francese, Italiano, Cinese, Portoghese, Brasiliano, Svedese e Greco. 2 milioni di lettori mensili. 140.000 visitatori web mensili. 43.000 abbonati a e-weekly, la newsletter digitale. Elektor propone progetti di elevata qualità ideali per l'autocostruzione nei campi dei microcontrollori, audio high-end, RC modelling, RF e test & measurement. Elektor è creata da veri appassionati di elettronica. Elektor è l'unica rivista internazionale indipendente a vantare un laboratorio di progettazione e PCB design interno.

### PERCHÉ SCEGLIERE TUTTE E TRE LE RIVISTE

Perché solo Inware Edizioni può darti una offerta così completa a condizioni così convenienti con un mix di aggiornamento, conoscenza, stimolo, divertimento e professionalità! Approfitta della fantastica PROMOTUTTE3 non te ne pentirai

CODICE MIP 2827175

di LEONARDO NERI

# UN ETILOMETRO con Arduino

**Realizziamo un semplice etilometro costituito da un sensore Mq-3 e la piattaforma Arduino. Uno strumento utile a evitare spiacevoli e pericolose situazioni.**

Dopo i sempre più frequenti incidenti provocati dall'uso eccessivo dell'alcool, siamo portati a prestare molta più attenzione a bere un bicchiere in più rispetto al passato. Infatti il codice della strada disciplina la normativa in questione con l'articolo 186 risalente al 1992 e recentemente inasprito. Per esempio basti pensare che un tasso alcolemico superiore a 0,5 g/l e inferiore 0,8 g/l può portare una sanzione pecuniaria che va da 500 a 2.000 euro e la sospensione della patente da tre a sei mesi.
Lo strumento necessario per la misura dell'alcool nel sangue è l'etilometro che, attraverso un trasduttore, converte la con-

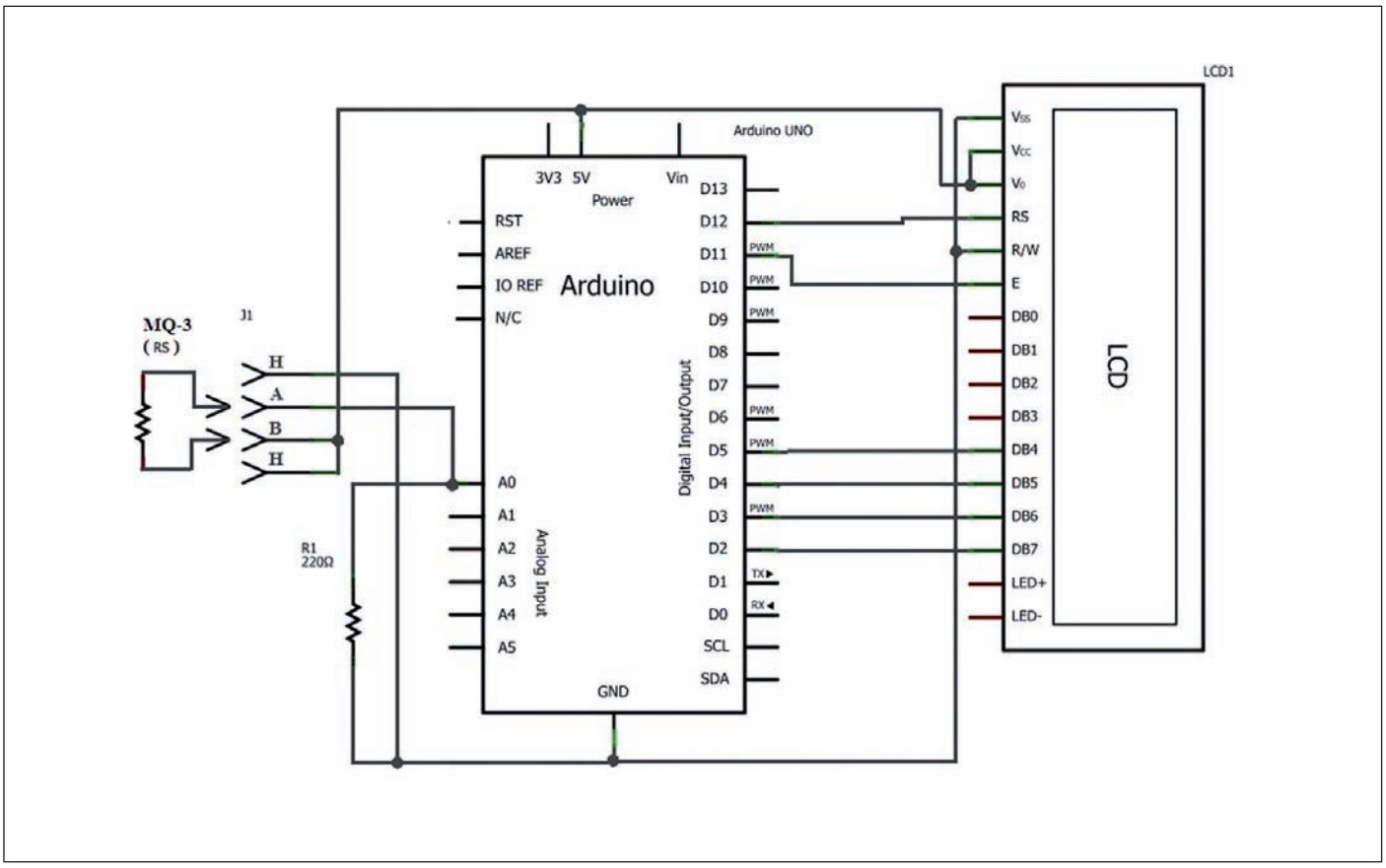



*Figura 3: schematico dell'etilometro-shield.*

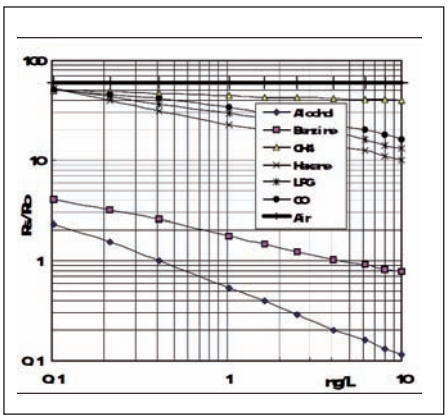



*Figura 1: Caratteristica di sensibilità.*

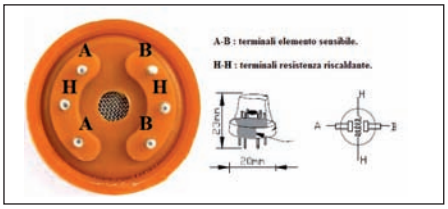



*Figura 2: pin out del sensore rilevatore di gas MQ-3.*

centrazione di alcool nel presente nel fiato di una persona in una grandezza elettrica. Il rapporto che lega la concentrazione di alcool nel respiro rispetto al sangue, in Italia è 2300:1. Perciò in questo progetto vogliamo presentarvi un semplice etilometro realizzato con un sensore Mq-3 e Arduino. Arduino rileva una tensione legata matematicamente alla concentrazione di alcol e, dopo averla processata, visualizza il tutto su un display LCD 16x2. Grazie alla semplicità e al basso costo è facilmente realizzabile da chiunque.

## IL SENSORE

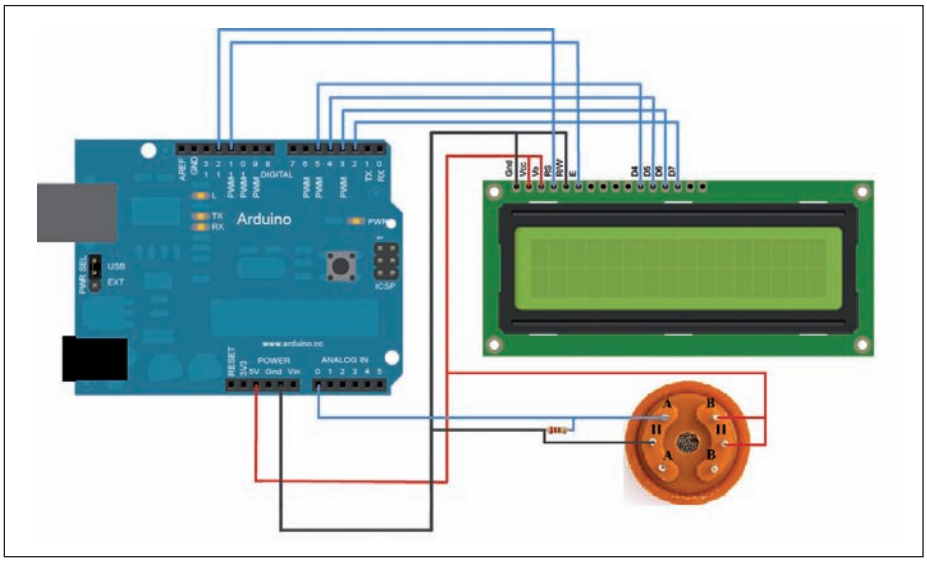
Il sensore in questione, MQ-3, è in grado di rilevare la concentrazione di alcool nell'aria da 0,5 mg/l a 10 mg/l fornendo in uscita un valore ohmico in funzione della concentrazione dell'alcool **(figura 1)**. La tensione di funzionamento è di 5 V e la sua resistenza a una concentrazione di 0,4 mg/l va da 4 Mohm a 8 Mohm.

Tale valore dipende da diversi fattori, quali: umidità, temperatura e persino dal fumo di sigaretta (10 sigarette sono rilevate dal sensore come una concentrazione di alcool pari a 0,1 mg/l). Per questo è importante un preciso sistema di calibratura, che esporremo più avanti. Al suo interno è presente una resistenza che permette di riscaldare l'elemento sensibile riconoscibile dai terminali denominati H **(figura 2)**. Per il suo funzionamento è necessario collegare al pin di uscita un resistore in modo da realizzare un partitore, utile per far acquisire ad Arduino un valore di tensione in funzione della concentrazione dei vapori di alcool. Sul manuale tecnico è consigliabile l'utilizzo di un trimmer da 200 kohm.

## HARDWARE

È interessante concepire questa realizzazione come un "etilometro-shield", ossia un'espansione della piattaforma Ar-

*Figura 4: schema di collegamento.*




duino che si interfaccia integralmente con i suoi connettori creando un sistema compatto e di facile espansione. L'alimentazione di 5 V necessaria al funzionamento della scheda è prelevata direttamente da Arduino. Il display LCD 16x2 è interfacciato tramite 6 pin, individuabili anche sulla libreria "LiquidCrystal": LCD RS sul pin 12, LCD EN sul pin 11, LCD D4 sul pin 5, LCD D5 sul pin 4, LCD D6 sul

pin 3, LCD D7 sul pin 2. Il sensore MQ-3 è stato cablato con un cavo multipolare terminando con un connettore strip maschio per facilitarne l'utilizzo. Ai capi dell'elemento riscaldante sono presenti 5 V, mentre in serie all'elemento sensibile è collegata R1 di 220 ohm verso GND. È importante misurare tale resistenza con un multimetro, giacché questo valore è determinante per il buon funzionamento dell'etilometro ed è un valore che in seguito utilizzeremo in sede di calcolo nel firmware (in questo prototipo misura 216 ohm). La tensione presenti ai capi di R1 viene elaborata da Arduino attraverso l'ingresso analogico A0 **(figura 3)**.

## FIRMWARE

Nella prima parte di codice Arduino attende che il sensore si riscaldi. Infatti finche il valore di tensione su R1 non si stabilizza e comunque, se il sensore misura un valore troppo altro, LCD mostra "ATTESA". Dopo questa fase di "calibrazione", viene letta la tensione ai capi di R1 e convertita dal convertitore ADC. Il firmware, in base al valore acquisito, calcola la corrente sul ramo R1-RS. Applicando semplicemente la legge di ohm ricava il valore ohmico del sensore (RS). Con il rapporto tra RS e la resistenza del sensore a 0,4 mg/l (RO) otteniamo un valore con cui possiamo ricavare la concentrazione di alcool semplicemente consultando il grafico delle caratteristiche del trasduttore.
A questo punto è importante calcolare l'equazione della retta di tale grafico: Y = – 069x – 0,276; l'incognita è la concentrazione di alcool quindi X = (0,276 + Y)/ – 0,69. Da notare che il grafico è in scala logaritmica perciò sulla formula è necessario inserire il $Log_{10}(Y)$ ossia il $Log_{10}$ del rapporto RS/RO e per ottenere la concentrazione di alcool è necessario eseguire l'esponenziale base 10 di X ($10^X$). Tali

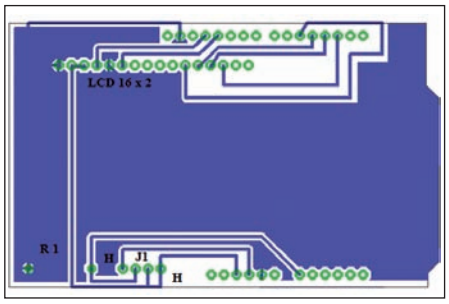


*Figura 5: PCB etilometro shield.*

calcoli, grazie alle funzionalità di Arduino, sono molto semplici da implementare sul software togliendoci gran parte del lavoro. Il valore così letto, al momento, rappresenta la reale concentrazione in mg/l sul respiro. La legislazione italiana impone che il rapporto tra concentrazione di alcool nel sangue e nel fiato sia 2.300:1 perciò il micro moltiplica il valore precedentemente ottenuto per 2.300 ottenendo così il rapporto g/l nel sangue.

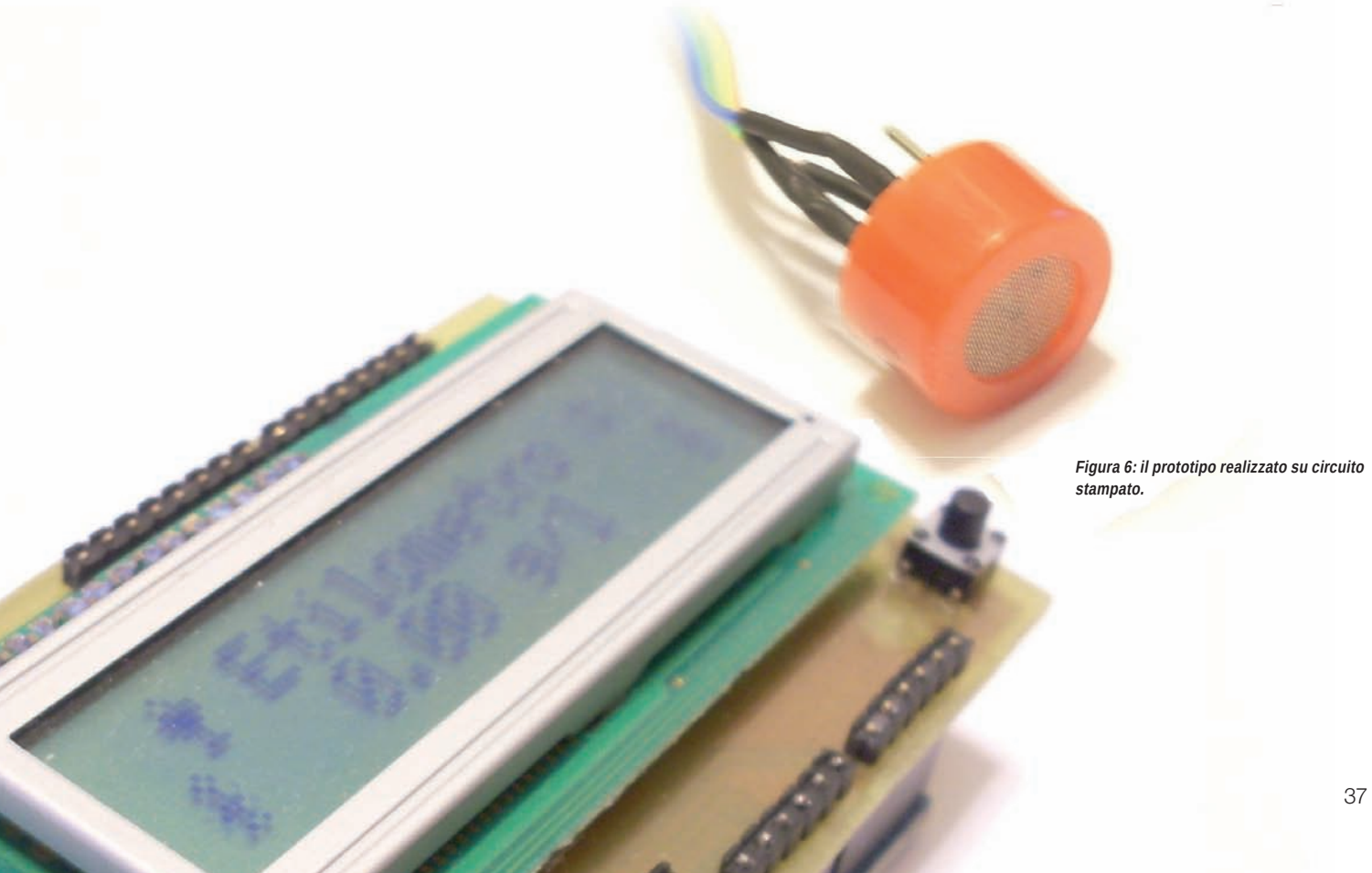


*Figura 6: il prototipo realizzato su circuito stampato.*

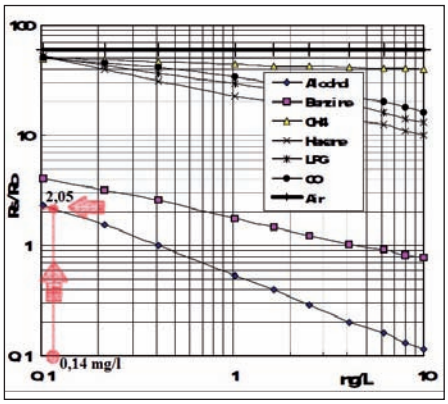



*Figura 7: ottenere Ro dal grafico in riferimento all'esempio.*

## CONCLUSIONI

Occorre sottolineare, prima di calibrare o eseguire semplici letture sul sensore, che è necessario attendere un periodo di "start-up" pari a 24-48 ore. Tale periodo è necessario per rendere stabile il valore ohmico dell'elemento sensibile. RO in questo caso è stata misurata in un laboratorio chimico a una concentrazione di alcool pari a 0,4 mg/l, ma risulta possibile misurare tale sensore anche dopo l'assunzione di una bevanda a basso contenuto alcoolico.

Avendo nota la gradazione e la quantità di bevanda assunta e grazie a numerosi tool reperibili in rete è possibile calcolare la contrazione di alcool nel sangue in base al sesso, peso e altezza del soggetto. Un esempio: dopo circa 30 minuti (in base alle condizioni fisiologiche del soggetto) dall'assunzione di una birra con 5% di alcool da 330 cl su un soggetto maschio di 75 kg per 180 cm di altezza sono presenti circa 0,32 g/l di alcool nel sangue. Dividendo ciò per 2300 otteniamo circa 0,14 mg/l di alcool nel respiro. Leggiamo la resistenza del sensore con un multimetro e con tale concentrazione, RS risulta essere di 10 kohm. Inserendo il valore di concentrazione nell'equazione della retta caratteristica osserviamo che RS/R0 = 2,05 quindi RO = 10 Kohm / 2,05 = 4878 ohm **(figura 7)**. Tale valore sarà inserito sul firmware. Ripetiamo: questo è solo un esempio a scopo prettamente informativo. Si sconsiglia peraltro di adottare tale sistema e quindi calibrare il sensore rivolgendosi a un laboratorio chimico.

Questo etilometro non può essere paragonato a dispositivi più costosi e sofisticati ma senza dubbio vi aiuterà a evitare spiacevoli situazioni. ❑

**CODICE MIP 2827034**

elettroshop.com
brilliant electronics since 1998

Utente anonimo... Login...
Il tuo carrello è vuoto

Home Page | Chi siamo | Listino/Info | Ordina | Download | Contatti | FAQ

CATEGORIE PRODOTTI

- Convertitori Serial-Ethernet
- Convertitori di protocollo
- Bluetooth
- Moduli radio e ZigBee
- Visualizzazione
- Modem GSM/GPRS/GPS
- Interfacce USB e PCI
- CPU e moduli
- Componentistica varia
- Sistemi di sviluppo HW/SW
- Programmatori
- CAE - CAD
- Strumentazione
- CD-ROM e pubblicazioni

RICERCA PRODOTTI
Vai
Ricerca avanzata

NEWSLETTER

Iscriviti alla nostra Newsletter (potrai sempre cancellarti in seguito)

Nome

Email

Conferma

## ARDUINO UNO

Scheda microcontrollore basata su ATMega328. Fornisce 14 pin con input/output digitali, 6 input analogici, un cristallo oscillatore a 16 MHz, una connessione USB, un jack di alimentazione, un header ICSP e un pulsante di reset.

€ 27,00 Info

## ARDUINO DUEMILANOVE

Piattaforma open-source basata su una semplice scheda I/O e su un ambiente di sviluppo user-friendly che implementa il linguaggio Processing/Wiring.

€ 23,00 Info

## ARDUINO BLUETOOTH

Versione Bluetooth della scheda Arduino. Il modulo Bluetooth è un modulo classe 1 (WT11) per una distanza di trasmissione di 100m.

€ 99,00 Info

## ETHERNET SHIELD

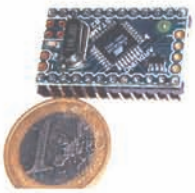
Grazie a Ethernet shield è possibile connettere una scheda Arduino a una LAN utilizzando la libreria Ethernet.

€ 31,00 Info

## ARDUINO NANO

Tutte le funzionalità della Duemilanove e più ingressi analogici con questa versione miniaturizzata della scheda Arduino.

€ 45,00 Info



## ARDUINO MINI

Versione miniaturizzata delle scheda Arduino (senza parte USB. Le dimensioni estremamente ridotte (30x18mm) permette una riduzione dello spazio di ingombro non indifferente.

€ 24,00 Info

CODICE MIP 2827179

I prezzi si intendono IVA esclusa

ARDUINO
lo trovi su
www.elettroshop.com

di MARCO CARMINATI

# SENSORI *di polveri* FINI

**Sempre più spesso le amministrazioni comunali ricorrono al blocco del traffico automobilistico nelle grandi città (nonostante i dubbi sull'efficacia di tali misure di contrasto dell'inquinamento), segnale della crescente gravità dell'impatto del particolato atmosferico sulla nostra salute. Approfondiamo quindi i principi di funzionamento dei vari sensori elettronici esistenti per misurare in tempo reale la concentrazione e le dimensioni delle polveri presenti nell'aria.**

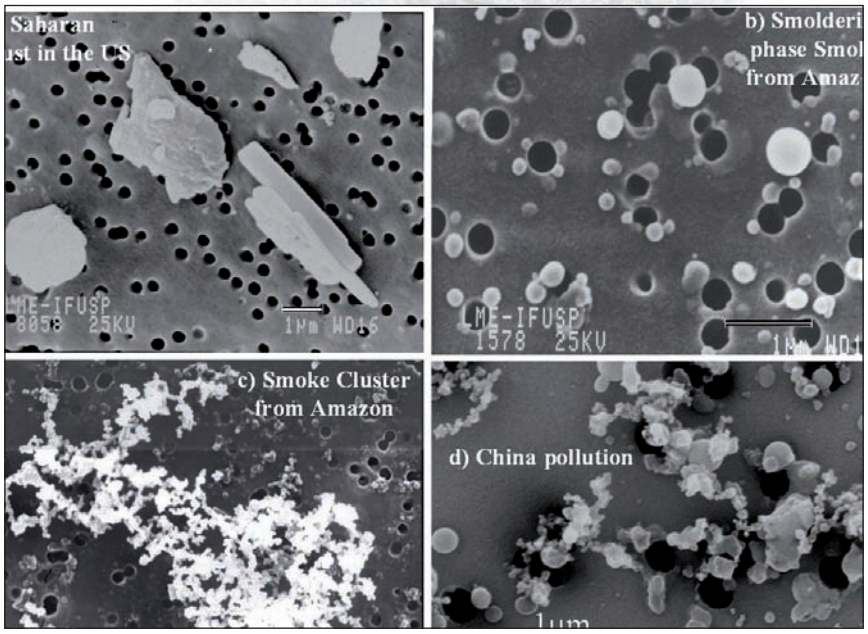



*Figura 1: esempi di particolato raccolto sui filtri in diverse aree geografiche. La barra di riferimento corrisponde a 1 µm (immagini al microscopio elettronico dal web).*

Le poveri fini, e in generale il particolato atmosferico, sono argomenti di drammatica attualità, rappresentando una delle componenti più nocive dell'inquinamento dell'aria delle nostre città. Infatti è ormai assodato che esiste una correlazione tra l'aumento della concentrazione del particolato e l'incremento dell'incidenza statistica di varie patologie nella popolazione urbana.
Tuttavia, i complessi meccanismi tossicologici scatenati dalla penetrazione del particolato nei polmoni, subdolo vettore microscopico di sostanze inquinanti, sono tuttora oggetto di studio. I sensori e gli strumenti elettronici in grado di misurare in tempo reale la concentrazione e la distribuzione delle polveri rappresentano quindi un'arma estremamente interessante per contrastare l'inquinamento, monitorando in modo sempre più accurato la qualità dell'aria e contribuendo all'identificazione dei legami causa-effetto e quindi alla messa a punto di mirate politiche di intervento.

## ORIGINE E CLASSIFICAZIONE

Il particolato è costituito da un insieme molto variegato di particelle solide e liquide disperse nell'atmosfera **(figura 1)**. Queste sono sia di origine naturale (eruzioni vulcaniche ed erosione) sia prodotte dall'attività umana. In particolare, nei contesti urbani la combustione (motori dei veicoli e riscaldamento degli edifici) rappresenta la causa principale dell'incremento significativo della concentrazione di polveri nell'atmosfera.

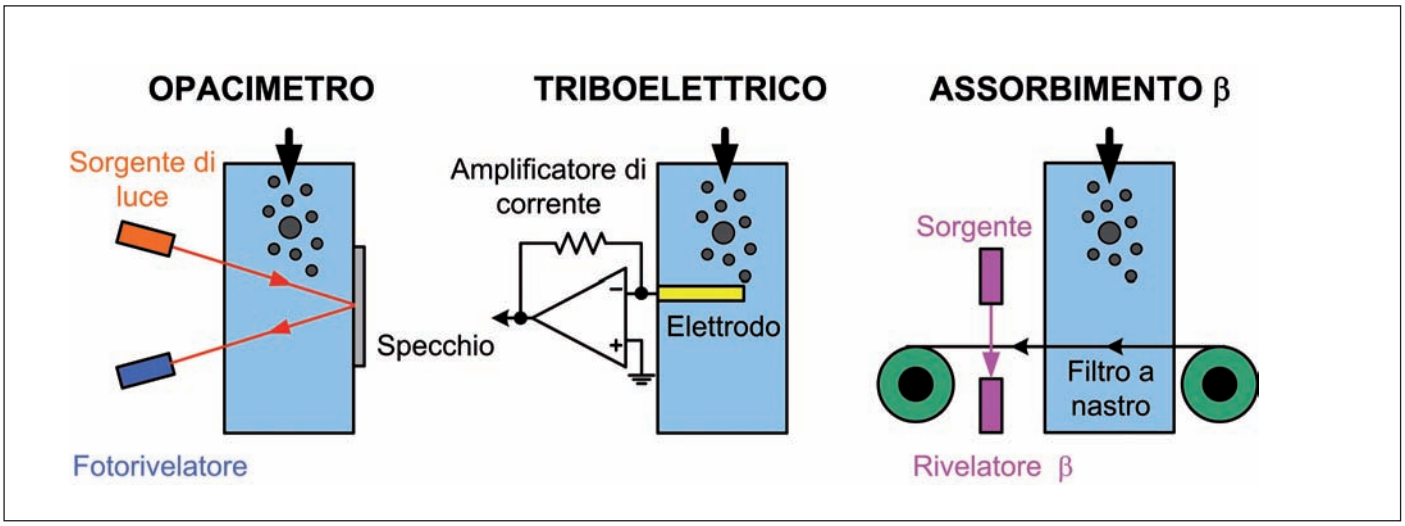



*Figura 2: sensori elettronici che consentono la misura della concentrazione di particolato in tempo reale.*

Il particolato viene classificato in base al diametro aerodinamico equivalente delle particelle (non essendo necessariamente sferiche). Le polveri con diametro compreso tra le decine di micron fino a 100 nm costituiscono il particolato fine, nel quale la classe più nota è il PM10 (inferiori a 10 μm). Le nanoparticelle sotto 100 nm costituiscono invece il particolato ultrafine.

## PRINCIPI DI MISURA

Il metodo tradizionale di misura della concentrazione delle polveri consiste nella loro raccolta su filtri che vengono accuratamente pesati in modo da risalire al peso delle particelle solide depositatesi, corrispondente a un certo volume d'aria fatto fluire attraverso il filtro per un tempo e con una portata definiti. Questa tecnica, detta gravimetrica, fornisce un'informazione sul peso complessivo delle polveri (espresso in μg/m$^3$). Sebbene sia l'unica tecnica attualmente riconosciuta a livello normativo, essa presenta gravi limiti tra cui la necessità di un operatore che segua un preciso protocollo di pesatura manuale e l'elevato tempo di campionamento necessario (alcune ore).
Sono state quindi sviluppate tecniche elettroniche alternative **(figura 2)** che permettono una misura della concentrazione delle polveri in tempo reale. Gli *opacimetri* sono sensori optoelettronici che misurano l'estinzione della luce, ovvero la riduzione dell'intensità di un fascio luminoso dovuta a fenomeni di assorbimento, riflessione e diffusione causati dalla presenza di particolato lungo il suo percorso.

I sensori *triboelettrici* misurano la corrente dovuta alle cariche prodotte elettrostaticamente dall'impatto del particolato che fluisce ad alta velocità contro un elettrodo metallico. Gli strumenti ad assorbimento misurano l'attenuazione di un fascio di raggi dovuta all'assorbimento del particolato raccolto su un filtro a nastro. Operano quindi in maniera analoga agli opacimetri, rimpiazzando però la radiazione luminosa con i raggi in modo da evitare l'incertezza dovuta alla dispersione delle proprietà ottiche del particolato.

## LASER SCATTERING

La limitazione fondamentale dei sensori sopra descritti risiede nell'incapacità di fornire un'informazione sulle dimensioni del particolato misurato. È assodato che dal punto di vista tossicologico la pericolosità delle particelle dipende dal loro diametro: quanto più sono piccole, tanto più in profondità penetrano nei polmoni, fino agli alveoli da dove si insinuano nel sistema circolatorio. È importante dunque conoscere non solo la concentrazione media delle poveri in peso (dominata dalle poche più grandi) ma soprattutto la loro distribuzione dimensionale.
L'unico strumento in grado di eseguire l'analisi delle dimensioni delle singole particelle (detta *granulometria*) è di tipo optoelettronico, basato sul fenomeno del *laser scattering*. Come illustrato in **figura 3**, l'aerosol aspirato dall'ambiente con un'opportuna sonda viene focalizzato aerodinamicamente tramite flussi d'aria laterali per far sì che le particelle scorrano allineate in fila indiana ad alta velocità. Nella camera di detection un raggio laser è focalizzato sul fascio in modo che, in presenza di una particella, la luce venga deviata generando un impulso luminoso di scattering che viene raccolto dal fotorivelatore, posto ortogonalmente rispetto al fascio incidente. Uno specchio permette di recuperare la luce diffusa in direzione opposta, mentre degli elementi assorbenti (le cosiddette trappole di luce) evitano un accoppiamento diretto tra il laser ed il fotodiodo.
Così, non solo è possibile misurare il numero di particelle che scorrono, contando il numero di impulsi acquisiti per un dato volume d'aria, ma è anche possibile ricavare una stima delle dimensioni di ogni singola particella dall'ampiezza dell'impulso, gra-

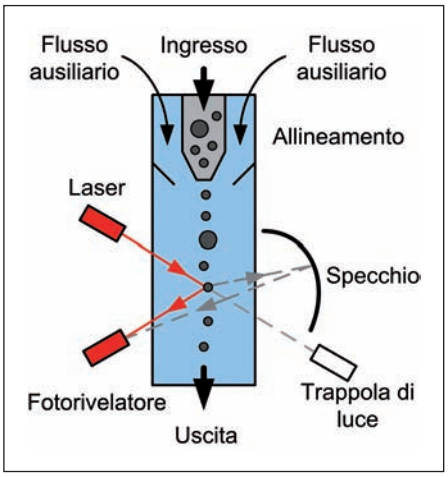



*Figura 3: analisi delle singole polveri per mezzo del laser scattering.*

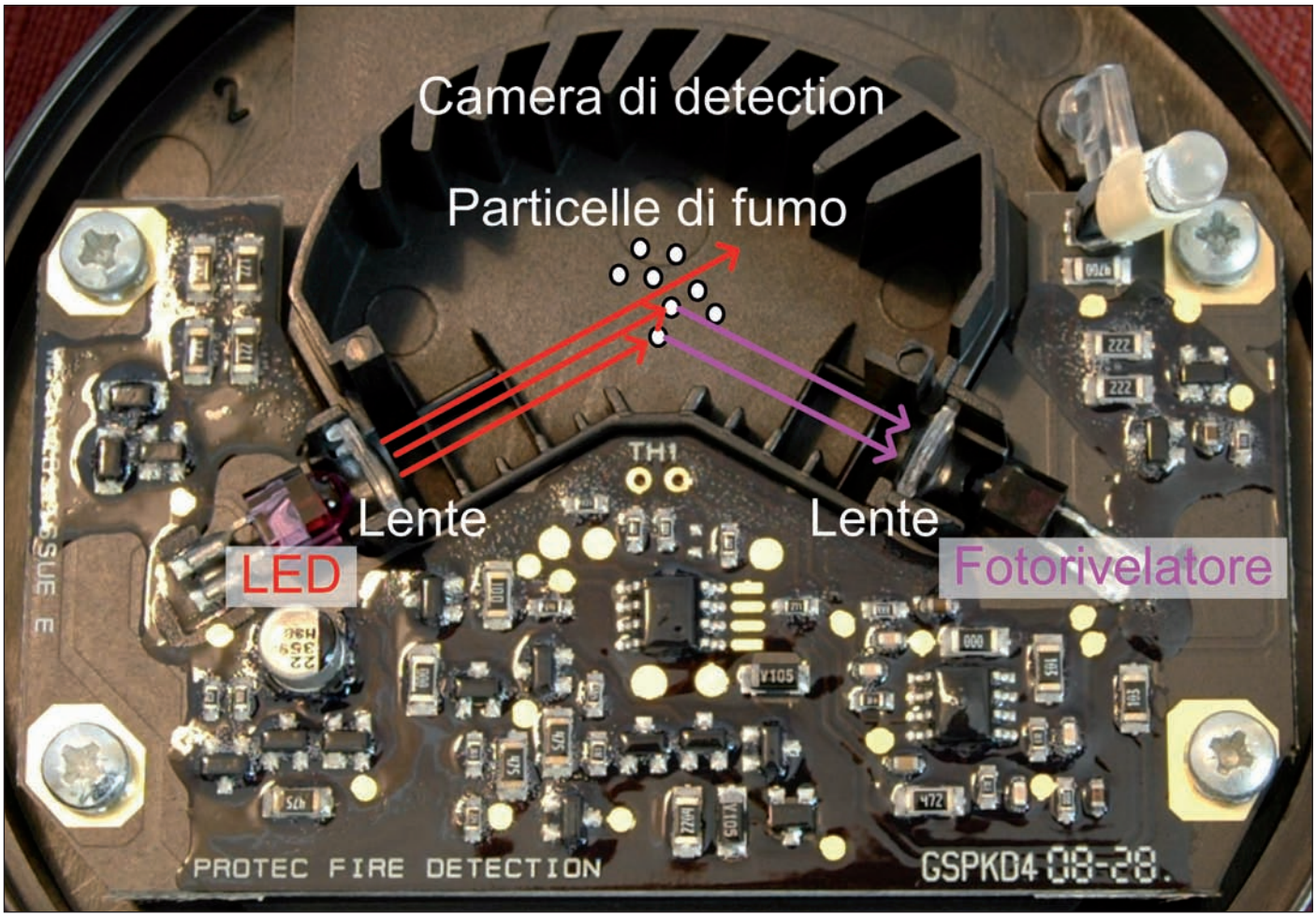



*Figura 4: sensore di fumo.*

zie a un legame ben definito esistente tra il diametro della particella (e le sue proprietà ottiche) e l'intensità dell'impulso luminoso (descritto dalla teoria di Gustav Mie). Questi sofisticati strumenti sono in grado di operare con concentrazioni tra 1 e $10^6$ particelle per litro, con diametri da circa 300 nm a 30 µm (suddivise in una decina di classi granulometriche).

Per rilevare nanoparticelle di diametro inferiore a 300 nm (limite dettato dalla lunghezza d'onda del laser, tipicamente nel rosso, e dalle proprietà dello scattering) esistono strumenti che aumentano artificialmente il diametro delle polveri, facendo condensare attorno vapori di alcol in modo da raggiungere un diametro della goccia condensata attorno al nucleo solido di circa 1 µm e quindi usare un contatore optoelettronico. Poiché in questo modo si perde l'informazione sulla dimensione originale delle particelle, queste vengono separate a monte con tecniche basate sulla mobilità elettrica, che dipende dal loro volume.

## SENSORI DI FUMO

Una sottoclasse di sensori di polveri molto più diffusa e abbordabile per lo sperimentatore elettronico è quella dei sensori di fumo, installati capillarmente negli edifici pubblici, nei classici contenitori circolari a soffitto, per la tempestiva rilevazione degli incendi. Anche in questo caso la presenza delle polveri è rilevata tramite lo scattering ottico, ma senza la necessità di conoscere le dimensioni delle singole particelle. Inoltre, la concentrazione di particolato molto elevata nel fumo (~10 µg/m$^3$) che si muove lentamente nel dispostivo permette di rilassare notevolmente il progetto del sensore, in cui il laser è tipicamente sostituito da un LED nell'infrarosso. Come illustrato in **figura 4** la camera di rivelazione è nera e ricca di lamelle plastiche che assorbono la luce evitando l'accoppiamento ottico diretto tra LED e fotodiodo.

Esistono circuiti integrati dedicati, come il RE46C165 di Microchip, che integrano tutti i blocchi circuitali necessari per gestire il sensore e richiedono solo qualche com-

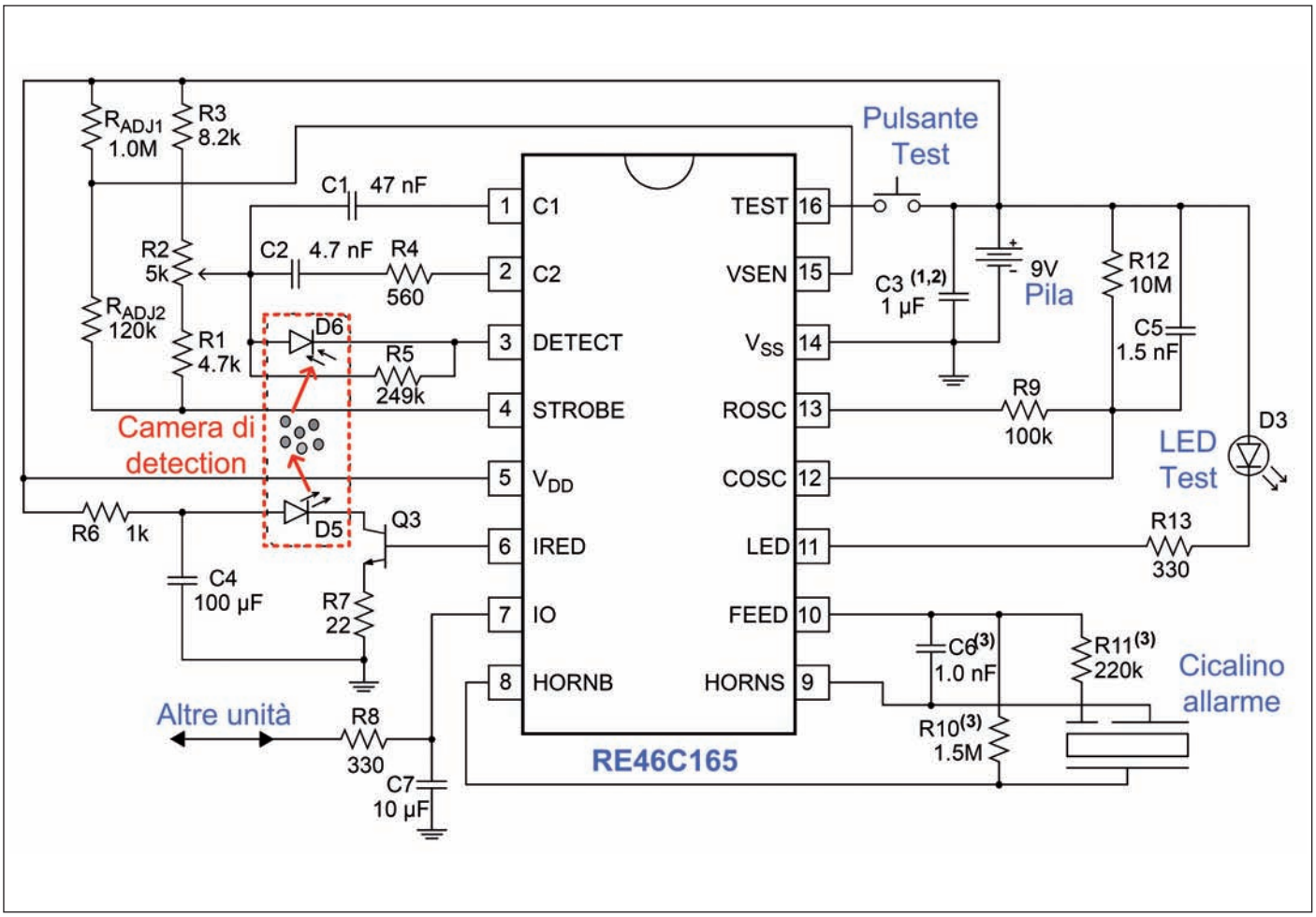



*Figura 5: schematico con la componentistica aggiuntiva richiesta dall'integrato RE46C165 per misurare lo scattering dovuto alle particelle di una nuvola di fumo.*

ponente esterno, come illustrato nello schema di **figura 5** tratto dal relativo datasheet. Il chip contiene il circuito di polarizzazione del LED e l'amplificatore che legge il segnale del fotodiodo.
Per minimizzare il consumo di potenza e allungare così la vita della batteria (corrente a riposo inferiore a 8 µA), l'integrato contiene anche un oscillatore interno che accende la sezione di detection per 100 µs ogni 10s. Se viene rivelato del fumo, la misura è ripetuta a maggiore frequenza per verificare la condizione d'allarme e attivare quindi il cicalino. Ogni 43 s viene controllato lo stato della batteria da 9 V e l'integrità della camera. Infine, una memoria consente di registrare l'avvenuta detection del fumo e una semplice logica digitale permette di eseguire il test del sensore alla pressione del relativo pulsante e di interfacciarlo con altri sensori tramite un'unica linea digitale (in modo da far suonare l'allarme di tutti i dispositivi in caso di incendio). ❑

**CODICE MIP 2827038**

## PER *approfondire...*


- G. Buonanno, F.C. Fuoco, L. Stabile, A. Viola. "La misura dell'esposizione alle polveri aerodisperse: dal PM10 alle nanoparticelle", Libellula Edizioni, 2011.
- W. C. Hinds, "Aerosol Technology", John Wiley and Sons, 1999.
- www.iasaerosol.it (Società Italiana Aerosol).

di GIOVANNI DI MARIA

Corso mikroBASIC

# La libreria PS/2

**In questo numero impareremo a utilizzare, in maniera approfondita, i display LCD testuali, soffermandoci sul modello fornito con la scheda di sviluppo EasyPIC**

## MOLTI TASTI

In passato Fare Elettronica ha pubblicato alcuni articoli che riguardavano il collegamento di alcuni pulsanti al microcontrollore PIC. Con un micro potente, e la predisposizione di un moltiplicatore di porte, si potrebbe arrivare a disporre di circa 40-50 ingressi. Utilizzando anche alcune tecniche (come ad esempio il multipartitore resistivo con ADC) si potrebbe anche disporre di un buon numero di input.
Tutte queste soluzioni però implicano l'utilizzo quasi totale di tutte le porte di comunicazione di una MCU, rimanendo pressoché sprovvisti di ulteriori ingressi o uscite per altri scopi.
Si è deciso quindi di cambiare strategia, dopo che l'autore ha ricevuto la richiesta, da parte di un cliente, di realizzare un dispositivo elettronico dotato di circa 90 tasti. Si immagini di eseguire il cablaggio di novanta pulsanti: un'operazione estremamente laboriosa ed esteticamente pesante.
C'era solamente una soluzione proponibile: quella di ricorrere a una tastiera PS/2, utilizzando solamente due porte del microcontrollore e riducendo al minimo le connessioni.

## IL COLLEGAMENTO

Collegare una tastiera PS/2 a un PIC è molto semplice. Sono sufficienti infatti due soli collegamenti per avere a portata di mano più di cento tasti. Lo schema elet-

*Figura 1: PIC e tastiera.*

trico di principio è quello mostrato in **figura 1**. Se si desidera, si può approntare un connettore femmina PS/2 da saldare direttamente su PCB. In questo modo si ha a disposizione un comodo metodo per collegare o rimuovere la tastiera, quando si desidera.

## KEYBOARD E EASYPIC

Per coloro che utilizzano la EasyPIC è importante far notare che sulla scheda è presente un connettore PS/2 per il collegamento di una tastiera o di un mouse. Per permettere e abilitare la connessione è necessario impostare correttamente i due jumper J20 e J21 che stabiliscono un collegamento tra le due linee DATA e CLK rispettivamente e le porte del PIC RC0 e RC1. Si consiglia di collegare o staccare la tastiera dal connettore PS/2 solo quando la EasyPIC è spenta, per scongiurare danni irreversibili al microcontrollore.

## LE FUNZIONI MIKROBASIC

Veniamo adesso alla carrellata, alquanto ridotta, delle funzioni e procedure messe a disposizione del mikroBasic, per la gestione completa della comunicazione con i dispositivi PS/2. La libreria è composta solamente da due routine e, a riguardo, occorre ricordare alcuni aspetti fondamentali:

- la libreria non utilizza alcun tipo di interrupt;
- la gestione della comunicazione PS/2 richiede un oscillatore (quarzo) con frequenza minima di 6 Mhz. Al bando quindi i quarzi da 4 Mhz;
- è consigliabile collegare le linee del bus PS/2 ad altrettante resistenze di pull-up;
- a differenza del collegamento tastiera con il Personal Computer, il sistema non prevede un colloquio "all'indietro" del segnale (MCU a tastiera). Questo vuol dire che se si preme il tasto CAPS LOCK sulla tastiera, il relativo diodo LED verde non si illuminerà. Tale comportamento e del tutto normale e non scambiatelo dunque per un malfunzionamento.

### Ps2_Config

Questa procedura serve per inizializzare e preparare il microcontrollore alla comunicazione con il bus PS/2. Non ha bisogno di parametri, ma il suo lavoro dipende da alcune variabili globali:

- PS2_Data: Data signal line;
- PS2_Clock: Clock signal line;
- PS2_Data_Direction: Direction of the Data pin;
- PS2_Clock_Direction: Direction of the Clock pin.

Queste variabili devono essere definite all'inizio del programma, tramite il comando DIM. Ad esempio:

```
dim PS2_Data as sbit at RC0_bit
dim PS2_Clock as sbit at RC1_bit
dim PS2_Data_Direction as sbit at
TRISC0_bit
dim PS2_Clock_Direction as sbit
at TRISC1_bit
```

Esempio di utilizzo:

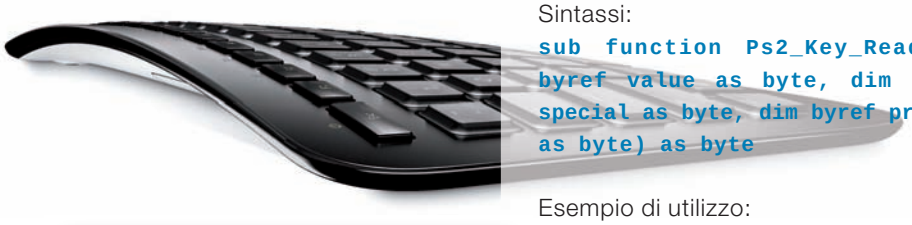
```
Ps2_Config()
```

### Ps2_Key_Read

È naturalmente la funzione più importante, quella che svolge il reale lavoro di comunicazione e ha il compito di recuperare le informazioni sul tasto pressato della tastiera. Essa ritorna un valore numerico: 0 se nessun tasto è stato premuto, 1 se la lettura dei tasti è avvenuta con successo.

La funzione accetta altresì tre parametri, passati per riferimento (indirizzo). Questo vuol dire che, in effetti, la funzione ritorna quattro valori, dal momento che gli altri tre sono modificati all'interno della funzione stessa.
I parametri passati sono i seguenti:

- *value*: contiene il valore di risposta del tasto premuto, ossia il codice ASCII;
- *special*: contiene un flag (0 o 1) a seconda che si sia premuto un tasto speciale (F1, Invio, Esc, ecc);
  - *pressed*: contiene un valore (0 o 1) indicante le pressione o meno di un tasto.



Sintassi:

```
sub function Ps2_Key_Read(dim byref value as byte, dim byref special as byte, dim byref pressed as byte) as byte
```

Esempio di utilizzo:

```
if (Ps2_Key_Read(value, special, pressed)) then
   if ((value = 13) and (special = 1)) then
      break
   end if
end if
```

TABELLA 1

| TASTO | VALORE RESTITUITO |
|---|---|
| F1 | 1 |
| F2 | 2 |
| F3 | 3 |
| F4 | 4 |
| F5 | 5 |
| F6 | 6 |
| F7 | 7 |
| F8 | 8 |
| F9 | 9 |
| F10 | 10 |
| F11 | 11 |
| F12 | 12 |
| Enter | 13 |
| Page Up | 14 |
| Page Down | 15 |
| Backspace | 16 |
| Insert | 17 |
| Delete | 18 |
| Windows | 19 |
| Ctrl | 20 |
| Shift | 21 |
| Alt | 22 |
| Print Screen | 23 |
| Pause | 24 |
| Caps Lock | 25 |
| End | 26 |
| Home | 27 |
| Scroll Lock | 28 |
| Num Lock | 29 |
| Left Arrow | 30 |
| Right Arrow | 31 |
| Up Arrow | 32 |
| Down Arrow | 33 |
| Escape | 34 |
| Tab | 35 |

*Tabella 1: valori dei tasti speciali*

## PRIMA ESPERIENZA: UN TASTO ACCENDE, UN ALTRO SPEGNE

Per prendere confidenza con il sistema, come prima esperienza, realizzeremo un semplice circuito a microcontrollore collegato a una tastiera PS/2. Il micro è dotato di un diodo LED collegato alla porta RB0. Se l'utente preme la lettera "A" della tastiera (maiuscola o minuscola) il diodo LED si illumina. Viceversa, se preme la lettera "Z" (maiuscola o minuscola) il diodo LED si spegne.

Per realizzare il programma, la prima co-

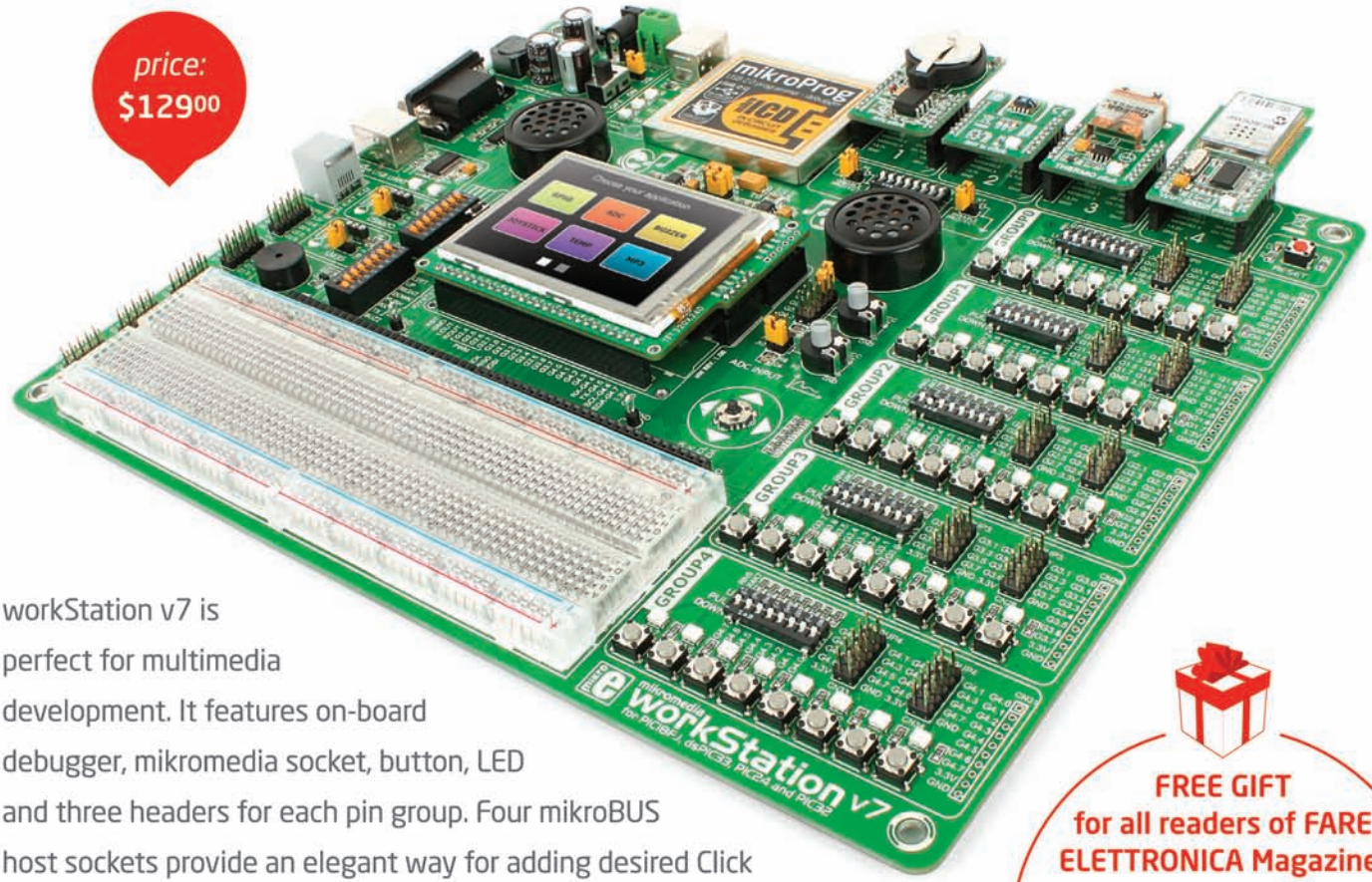
LISTATO 1

```
rem ********************************
rem *          Listato 1           *
rem *     MCU e Tastiera PS/2      *
rem *    Tasto 'A' accende Led     *
rem *    Tasto 'Z' spegne Led      *
rem *                              *
rem *    by Giovanni Di Maria      *
rem ********************************

program PS2_Example

dim keydata, special, down as byte

dim PS2_Data          as sbit at RC0_bit
    PS2_Clock         as sbit at RC1_bit
    PS2_Data_Direction  as sbit at TRISC0_bit
    PS2_Clock_Direction as sbit at TRISC1_bit

main:
  Ps2_Config()
  trisb=0
  portb=0
  Delay_ms(100)

  while TRUE
    if Ps2_Key_Read(keydata, special, down) then
        select case keydata
               case 65
                    portb.0=1
               case 97
                    portb.0=1
               case 90
                    portb.0=0
               case 122
                    portb.0=0
        end select
    end if
    Delay_ms(1)
  wend
end.
```

# mikromedia workStation v7

for PIC18FJ®, dsPIC33®, PIC24®, and PIC32®

price: $129.00




workStation v7 is perfect for multimedia development. It features on-board debugger, mikromedia socket, button, LED and three headers for each pin group. Four mikroBUS host sockets provide an elegant way for adding desired Click boards. Use large breadboard area to assemble your custom electronics. This miracle board is fantastic for education as well. We made sure to provide a rich set of libraries and examples for our mikroC, mikroBasic and mikroPascal compilers which facilitate development.

**FREE GIFT for all readers of FARE ELETTRONICA Magazine!**

**FARE-09-2012**

At the end of Checkout page enter the code above to GET FREE Shipping!

*ALL ORDERS INCLUDED

Code valid until 14th September 2012.

CODICE MIP 2814583

## PIC18FJ, dsPIC33, PIC24 or PIC32?

workStation v7 supports all mikromedia boards with Microchip® microcontrollers. Easily switch to the one you need.

**MikroElektronika**
DEVELOPMENT TOOLS | COMPILERS | BOOKS

www.libstock.com

**GET IT NOW**
www.mikroe.com

** Main picture includes: mikromedia workStation v7, mikromedia for PIC32, WiFi PLUS click, THERMO click, RTC2 click and SHT11 click.*

*** mikromedia and click boards are sold separately!*

*Figura 2: schema elettrico di connessione PS/2.*

sa da analizzare e da considerare è che occorre prevedere la pressione dei tasti maiuscoli e minuscoli. Questo perché l'utente potrebbe avere attivato o meno il tasto Caps Lock di cui, come detto prima, non ottiene la testimonianza dello stato logico, per via del relativo LED sempre spento.
I codici ASCII dei tasti previsti sono dunque i seguenti:

- lettera "A" = 65
- lettera "a" = 97
- lettera "Z" = 90
- lettera "z" = 122

Si tratta quindi di implementare, all'interno di un ciclo infinito, la lettura in continuo della tastiera e l'esecuzione di un'azione, a seconda dei tasti premuti. Se si premono i tasti con codice ASCII

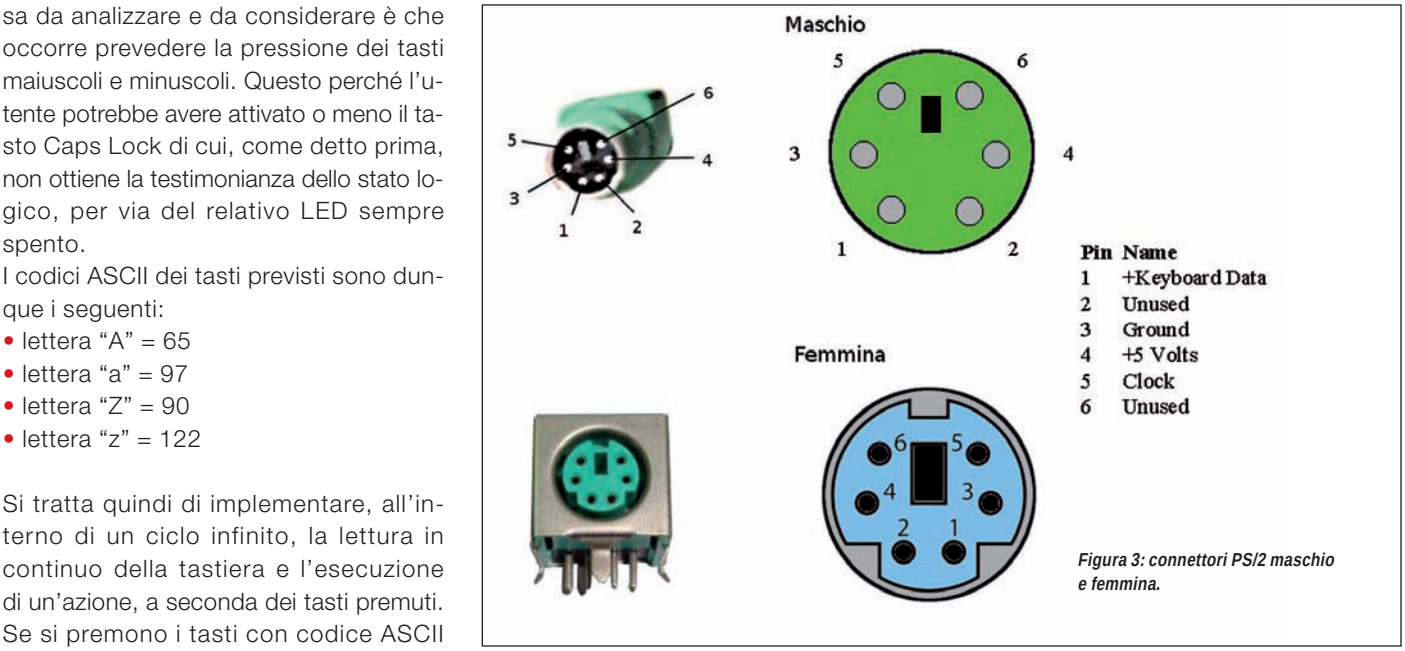



*Figura 3: connettori PS/2 maschio e femmina.*


48

## LISTATO 2

```
rem ******************************
rem *          Listato 2          *
rem *     MCU e Tastiera PS/2     *
rem *          Password           *
rem *                             *
rem *    by Giovanni Di Maria     *
rem ******************************

program PS2_Example

dim keydata, special, down as byte
dim stringa as string[20]
dim indice as byte

dim PS2_Data          as sbit at RC0_bit
    PS2_Clock         as sbit at RC1_bit
    PS2_Data_Direction  as sbit at TRISC0_bit
    PS2_Clock_Direction as sbit at TRISC1_bit

main:
  Ps2_Config()
  trisb=0
  portb=0
  Delay_ms(100)

  stringa=""
  indice=0

  while TRUE
     if Ps2_Key_Read(keydata, special, down)
then
       rem —-Se si digita MAIUSCOLA, la si ac-
coda alla stringa—-
       if (keydata >= 65) and (keydata <=90)
and (down=1) then
          stringa[indice]=  keydata
          indice=indice+1
       end if
       rem —-Se si digita minuscola, la si ac-
coda alla stringa come MAIUSCOLA—-
       if (keydata >= 97) and (keydata <=122)
and (down=1) then
          stringa[indice]=  keydata-32
          indice=indice+1
       end if
       rem —-Se si digita INVIO, esce——
       if keydata = 13 then
          break
       end if
    end if
    Delay_ms(1)
      rem ——-Se la stringa eccede la sua
lunghezza——-
    if indice>=19 then
       break
    end if
  wend
  rem ——-Inserisce il NULL nella stringa——-
  stringa[indice] =  0

  rem —-Se si indovina la Password————
  if strcmp( stringa , "FARELETTRONICA") = 0
then
     portb.0=1
  rem —-Se si sbaglia la Password————
  else
     while TRUE
        portb.0=1
        delay_ms(100)
        portb.0=0
        delay_ms(100)
     wend
  end if

end.
```

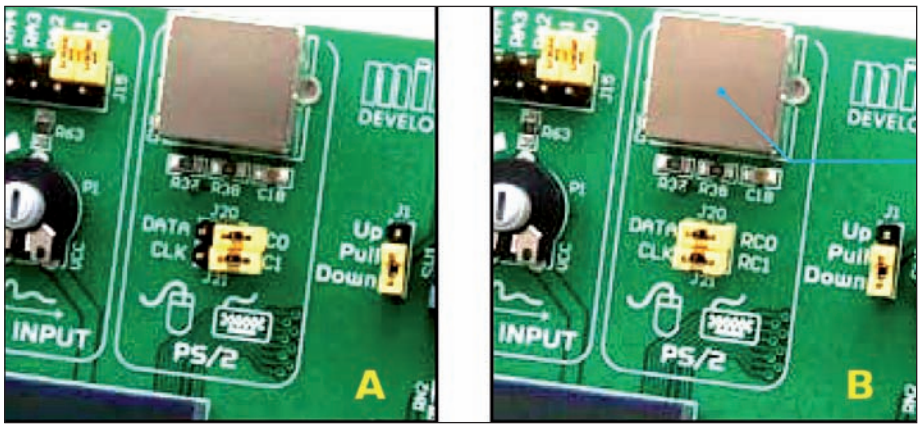



*Figura 4: PS/2 sulla EasyPic: a) J20 e J21 non connessi; b) J20 e J21 connessi.*

65 o 97 ("A", "a"), il diodo si deve accendere. Se si premono i tasti con codice ASCII 90 o 122 ("Z", "z"), il diodo si deve spegnere.
La scelta è affidata al blocco decisionale "select case". Tale blocco viene "letto" solo quando un tasto della tastiera è stato premuto **(listato 1)**.

È molto soddisfacente ed emozionante premere un tasto su una tastiera PS/2 e constatare che il microcontrollore a essa collegato risponde come previsto.

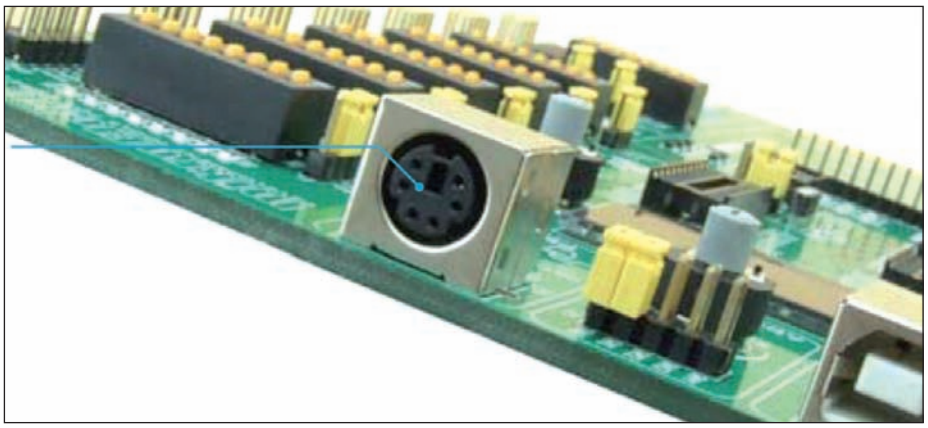


Figura 5: il connettore PS/2 sulla EasyPic.

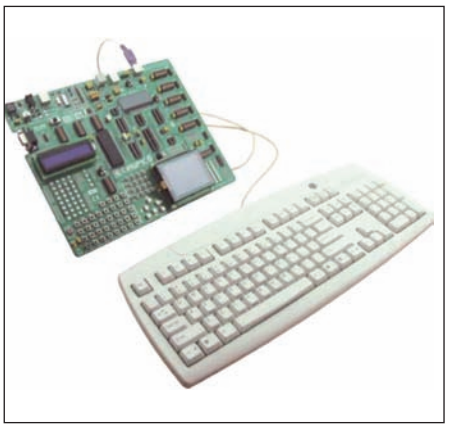


Figura 6: tastiera connessa alla EasyPic.

## SECONDA ESPERIENZA: PASSWORD CHE ATTIVA MCU

La seconda esperienza è più interessante e un poco più complicata. Si tratta di un sistema a microcontrollore, dotato di tastiera PS/2 e del solito diodo LED collegato sulla porta RB0. Il sistema, per essere attivato, ha bisogno della digitazione di una password. Se essa è corretta, il LED si illumina in modo fisso, confermando il buon esito dell'operazione e l'ingresso nel sistema da parte dell'operatore. Se la password è invece errata, il LED lampeggia velocemente, attestando lo stato di errore. La password corretta è FARELETTRONICA. Si può modificare la password da indovinare direttamente sul listato. Lo schema elettrico è il medesimo di quello utilizzato nella prima esperienza. L'operatore, dopo accensione del circuito, deve digitare sulla tastiera una password, senza curarsi dello stato in cui si trova il tasto CAPS LOCK. Alla fine dell'immissione del codice segreto, egli deve premere il tasto INVIO, per confermare. Il programma considera solo i tasti corrispondenti alla lettere dell'alfabeto. Qualsiasi altro carattere è ignorato. Il sistema inoltre dispone di una procedura che trasforma le lettere minuscole in maiuscole, uniformando in questo modo il controllo **(listato 2)**.

## CONCLUSIONI

Abbiamo aggiunto un tassello importante alla conoscenza dei microcontrollori. Le applicazioni che prevedono l'uso di una tastiera sono svariate. Applicazioni mobili e non, che prevedano l'uso di parecchi tasti, senza scomodare le preziose porte di una MCU. Senza contare poi che i prezzi di queste periferiche di input sono oggi estremamente bassi. Si possono anche utilizzare le piccole tastiere numeriche, dotate naturalmente di spinotto PS/2. Si eseguano tutte le prove del caso, utilizzando anche i tasti speciali.
Alle prossime puntate. ❑

**CODICE MIP 2827042**

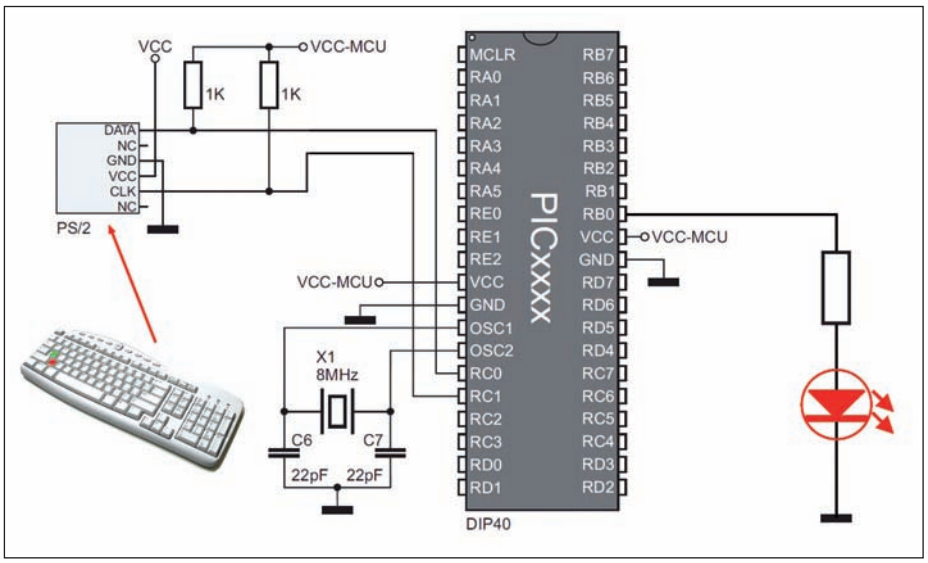



Figura 7: schema elettrico per la prima esperienza.

Figura 8: seconda esperienza: implementazione di una password.

Figura 9: una piccola tastiera numerica PS/2.

elettroshop.com
brilliant electronics since 1998

Utente anonimo... Login...
Il tuo carrello è vuoto

Home Page | Chi siamo | Listino/Info | Ordina | Download | Contatti | FAQ

CATEGORIE PRODOTTI

- Convertitori Serial-Ethernet
- Convertitori di protocollo
- Bluetooth
- Moduli radio e ZigBee
- Visualizzazione
- Modem GSM/GPRS/GPS
- Interfacce USB e PCI
- CPU e moduli
- Componentistica varia
- Sistemi di sviluppo HW/SW
- Programmatori
- CAE - CAD
- Strumentazione
- CD-ROM e pubblicazioni

RICERCA PRODOTTI
Vai
Ricerca avanzata

NEWSLETTER

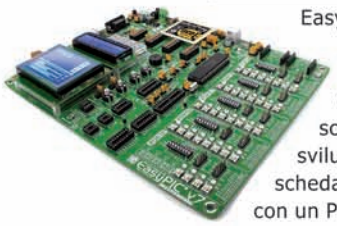
Iscriviti alla nostra Newsletter (potrai sempre cancellarti in seguito)

Nome

Email

Conferma

## EASYPIC7



EasyPIC v7 è la settima generazione della famosa scheda di sviluppo PIC. La scheda viene fornita con un PIC18F45K22.

€ 145,00 Info

## MIKROPROG

Programmer e In-Circuit Debugger per microcontrollori Microchip PIC, dsPIC e PIC32.

€ 89,00 Info

## SMARTGLCD

mikroElektronikaSmart GLCD 240x128 è uno strumento completo per lo sviluppo di applicazioni grafiche embedded utilizzando il PIC18F8722 connesso a un circuito oscillatore a 8MHz.

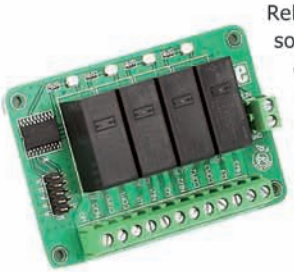
€ 79,00 Info

## MIKROBASIC-PRO-D

mikroBasic PRO for PIC è un compilatore BASIC completo per i microcontrollori Microchip PIC12, PIC16 e PIC18. Versione con Dongle USB.

€ 129,00 Info

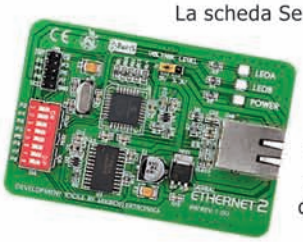
## EP-RELAY4 BOARD



Relay 4 Board è una soluzione economica e plug-n-play per aggiungere dei relè al vostro dispositivo.

€ 24,90 Info

## EP-S-ETHERNET2



La scheda Serial Ethernet 2 è uno strumento di sviluppo per il modulo Ethernet ENC624J600 di Microchip

€ 33,00 Info

CODICE MIP 2827187

I prezzi si intendono IVA esclusa

Nuovi tools di sviluppo
MikroElektronika
www.elettroshop.com

di MASSIMO BARBAGALLO

# MULTIKERNAL per Commodore 64

**Nei periodi gloriosi degli 8 bit per gli utenti del C64 era pratica diffusa sostituire il kernal per ottenere prestazioni superiori. Oggi, grazie a due semplici circuiti e pochi componenti, è possibile selezionare ben otto diversi kernal con la semplice pressione di un tasto.**

Il kernal del Commodore 64 "prima serie" è contenuto su una rom da 8 kbyte siglata MOS901227. Esso contiene una serie di routine volte alla gestione del I/O delle periferiche come, ad esempio, video, audio, tastiera ecc. Alcuni produttori di terze parti svilupparono kernal sostitutivi favorendo prestazioni e funzioni aggiuntive come, ad esempio, velocizzare il caricamento dati dal drive, funzioni di freezer e, in alcuni casi, l'accesso diretto al "Monitor".

Come detto sopra, il kernal era memorizzato su rom da 8 kbyte quindi, per la sostituzione, veniva fornita una EEPROM opportunamente pre-programmata. In al-

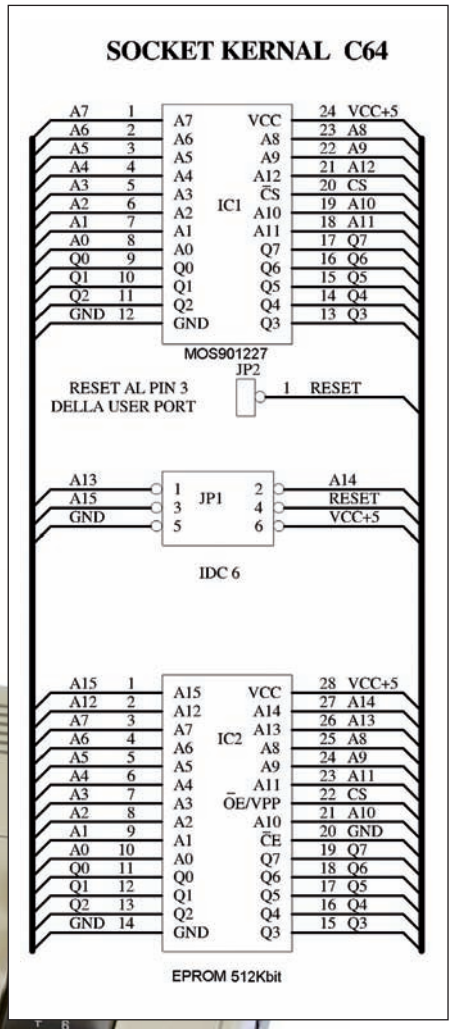



*Figura 1: prima sezione del multikernal.*

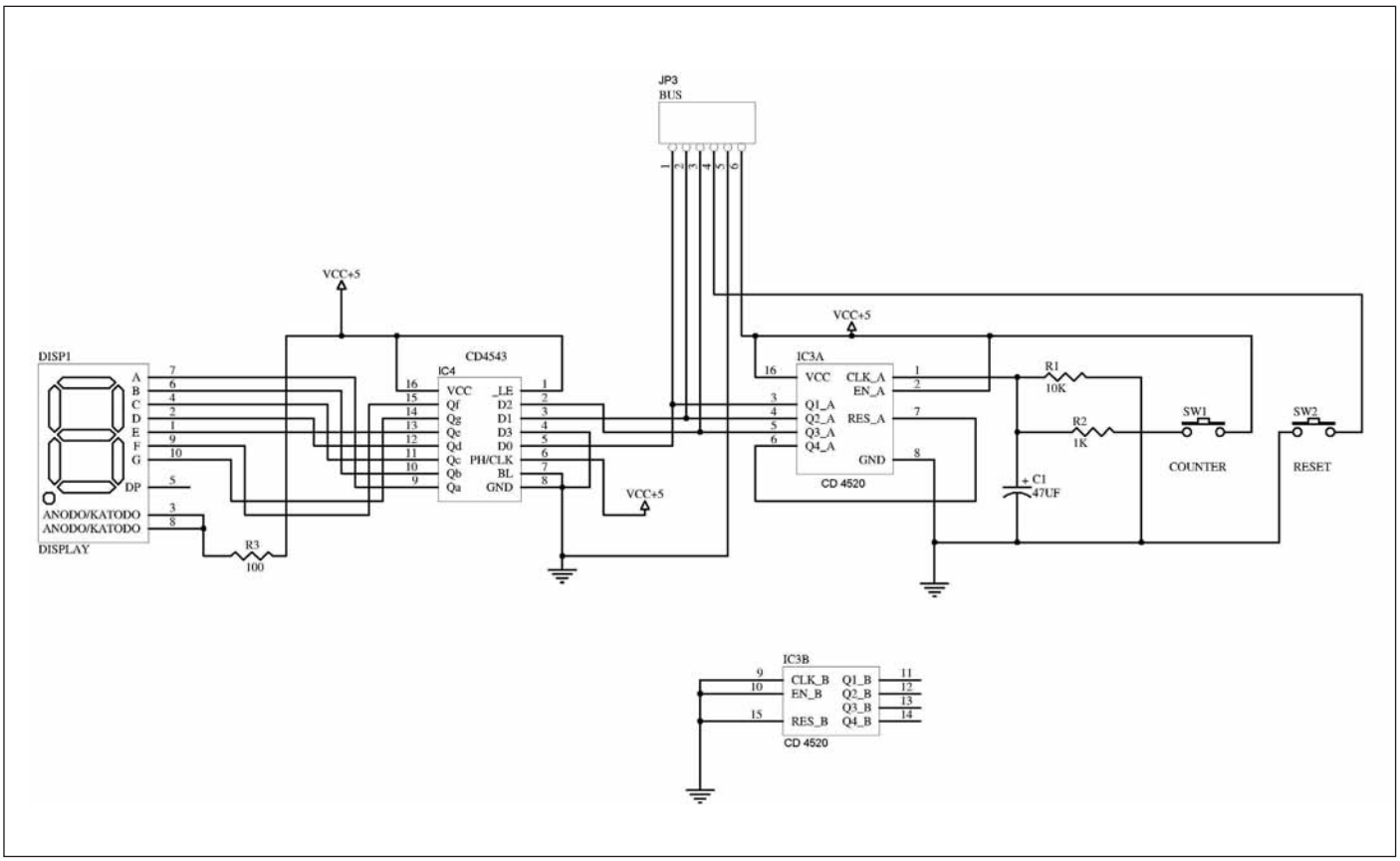



*Figura 2: telecomando.*

cuni casi veniva fornita solo una EPROM da 8 kbyte, così da contenere il solo kernal modificato perdendo, quindi, il kernal originale; in altri casi, con una EEPROM da 16 kbyte, venivano memorizzati due kernal: l'originale e quello modificato, che venivano selezionati per mezzo di un interruttore.

Ma non esisteva un solo kernal modificato: come oggi accade per i sistemi operativi, anche all'epoca, esistevano kernal di diverse case produttrici, ponendo in alcuni casi l'utente di fronte alla difficile scelta di quale installare. La sostituzione del kernal era relativamente semplice ma poco pratica. Essa consisteva nello smontare il C64, estrarre la rom originale "MOS901-227" e, quindi, installare la nuova rom. Dunque è evidente che la pratica di sostituzione non era tra le più comode. Da questa scarsa comodità nella sostituzione del kernal è nata l'idea di sviluppare un multikernal con il quale, adesso, è possibile selezionare otto o più kernal differenti tramite un tasto di selezione e, con un comodo telecomando munito di display, visualizzare la posizione del kernal e, in aggiunta, un pulsante di reset.

| Kernal | Decimale | Esadecimale | A15 | A14 | A13 |
|---|---|---|---|---|---|
| Originale | 0 - 8192 | $0000 - $1FFF | 0 | 0 | 0 |
| Sostituto 1 | 8192 - 16383 | $2000 - $3FFF | 0 | 0 | 1 |
| Sostituto 2 | 16384 - 24575 | $4000 - $5FFF | 0 | 1 | 0 |
| Sostituto 3 | 24576 - 32767 | $6000 - $7FFF | 0 | 1 | 1 |
| Sostituto 4 | 32768 - 40959 | $8000 - $9FFF | 1 | 0 | 0 |
| Sostituto 5 | 40960 - 49151 | $A000 - $BFFF | 1 | 0 | 1 |
| Sostituto 6 | 49152 - 57343 | $C000 - $DFFF | 1 | 1 | 0 |
| Sostituto 7 | 57344 - 65535 | $E000 - $FFFF | 1 | 1 | 1 |

*Figura 3: mappatura EPROM.*

Prima di entrare nel vivo della descrizione dello schema elettrico, è d'obbligo una breve delucidazione in merito alla rom MOS901227. Si tratta di una ROM (read only memory) da 8 kByte, quindi di sola lettura e, dunque, sono assenti i pin che la abilitano per la programmazione. Per poter inserire diversi kernal nella stessa memoria (non volatile) bisogna utilizzare una EPROM o una EEPROM. La differenza sostanziale tra le due sta nel modo di

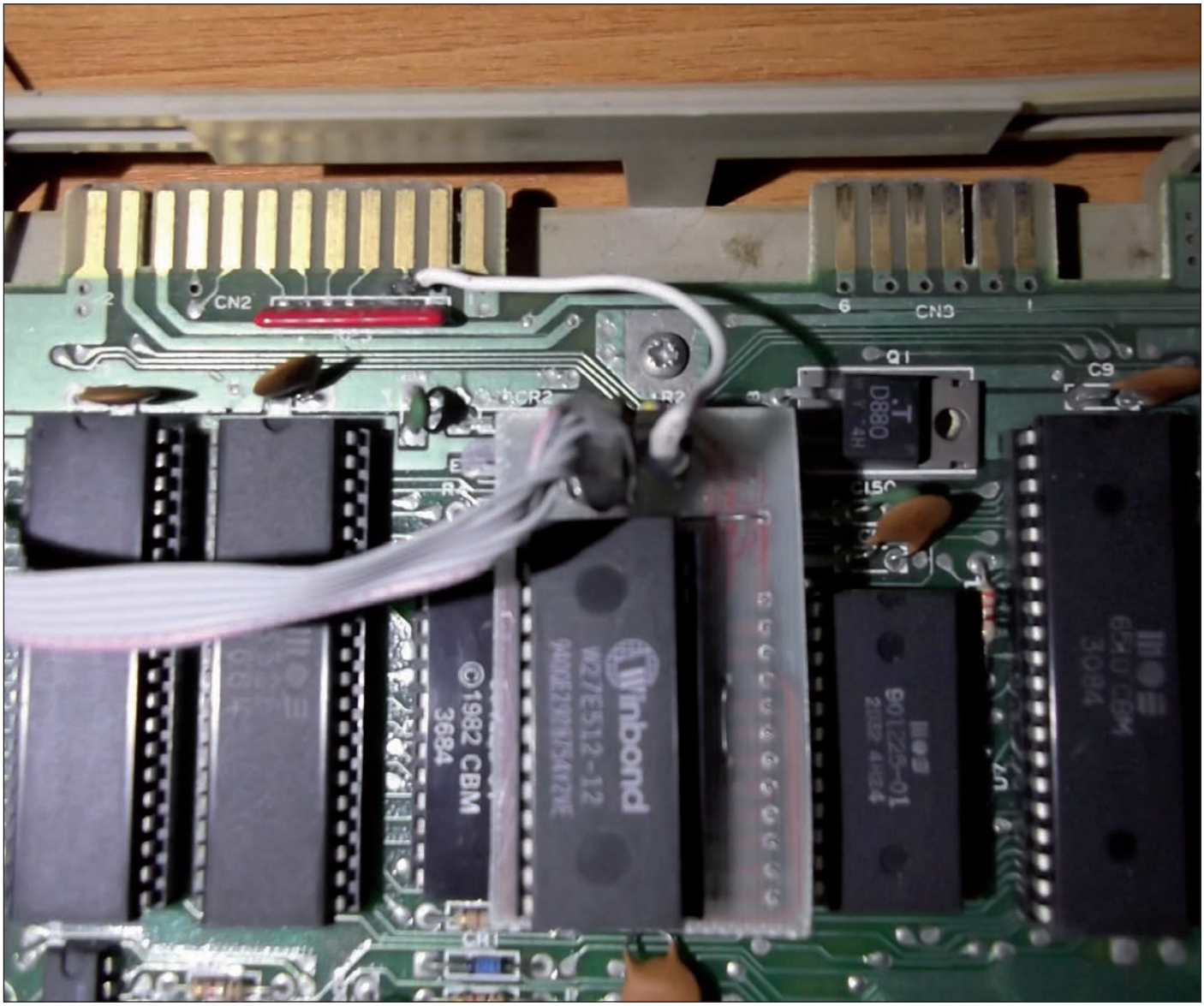



*Figura 4: l'adattatore EPROM installato al posto della ROM originale.*

LISTA COMPONENTI

| | |
|---|---|
| R1 | 10 kΩ 1/4 W |
| R2 | 1 kΩ 1/4 W |
| R3 | 100 Ω 1/4 W |
| C1 | 47 µF 16 V elettrolitico |
| SW1 | PULSANTE DA CS |
| SW2 | PULSANTA DA CS |
| DISP1 | DISPLAY 7 SEGMENTI ANODO |
| COMUNE JP1 | DOPPIO STRIP LINE 6 POLI |
| JP2 | STRIP LINE 1 POLO |
| IC1 | STRIP LINE 12 PIN X 2 |
| IC2 | EPROM DA 512 KBIT |
| IC3 | CD 4520 |
| IC4 | CD 4543 |

cancellazione: la prima tramite esposizione ai raggi UV, la seconda tramite conduzione elettrica. Nel nostro prototipo si è utilizzata una EEPROM W27E512-12.

Il kernal originale ha una dimensione di 8 kbyte, i sostituti devono rispettare esattamente questa grandezza fisica. Utilizzando una EPROM da 512 kbit abbiamo a disposizione 64 kbyte, quindi la possibilità di ospitare fino a otto kernal. Il MOS901227 e la W27E512-12 non hanno una piedinatura perfettamente uguale, quindi, serve riadattare le posizioni dei pin della W27E512-12 per essere alloggiata nel socket predisposto per il MOS901227.

## LO SCHEMA ELETTRICO

Il progetto del multikernal è costituito da due circuiti:

- alloggio della EPROM 512Kbit nel socket del MOS901227;
- telecomando.

Osservando lo schema elettrico del primo circuito visibile in **figura 1** notiamo ic1 e ic2; la nostra EPROM sarà installata su ic2 mentre ic1 è lo spazio dedicato

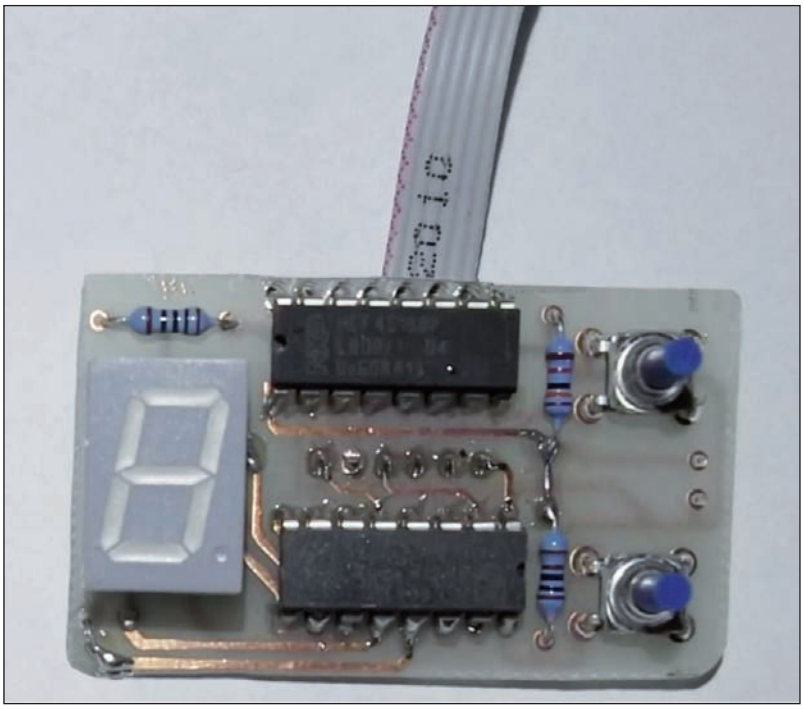


*Figura 5: il telecomando cavetto flat a 6 poli utilizzato per connetterlo all'adattatore.*

alla connessione con il socket. Le linee di indirizzamento A0 – A12 (13 bit) sono direttamente collegate al socket (ic1) così come il bus dati Q0 – Q7 (8 bit). Il pin cs (chip select) serve a forzare il bus dati (Q0 – Q7) a porsi in una condizione di alta impedenza liberando così il bus del C64 per svolgere altre funzioni, ad esempio, quando il processore interpella la rom basic o altri dispositivi e viene collegato a ic2 al pin OE (output enable), mentre il pin CE (chip enable) viene posto direttamente a massa (stato logico zero) così da tenere ic2 sempre attivo.
Abbiamo così riadattato i pin della EPROM al socket del C64; adesso ci occupiamo delle linee di comando per la gestione dei vari kernal. Per gestire gli otto kernal dobbiamo controllare le linee A13 – A15 che, come vediamo dallo schema in **figura 1**, sono collegate a idc6 il quale permette il collegamento con il telecomando tramite un cavo flat a 6 poli. Inoltre è presente un pin (jp2) che andrà collegato al pin 3 della userport del C64: si tratta del reset. Sempre al connettore IDC6 sono connessi Gnd e Vcc+5 che andranno ad alimentare il telecomando.

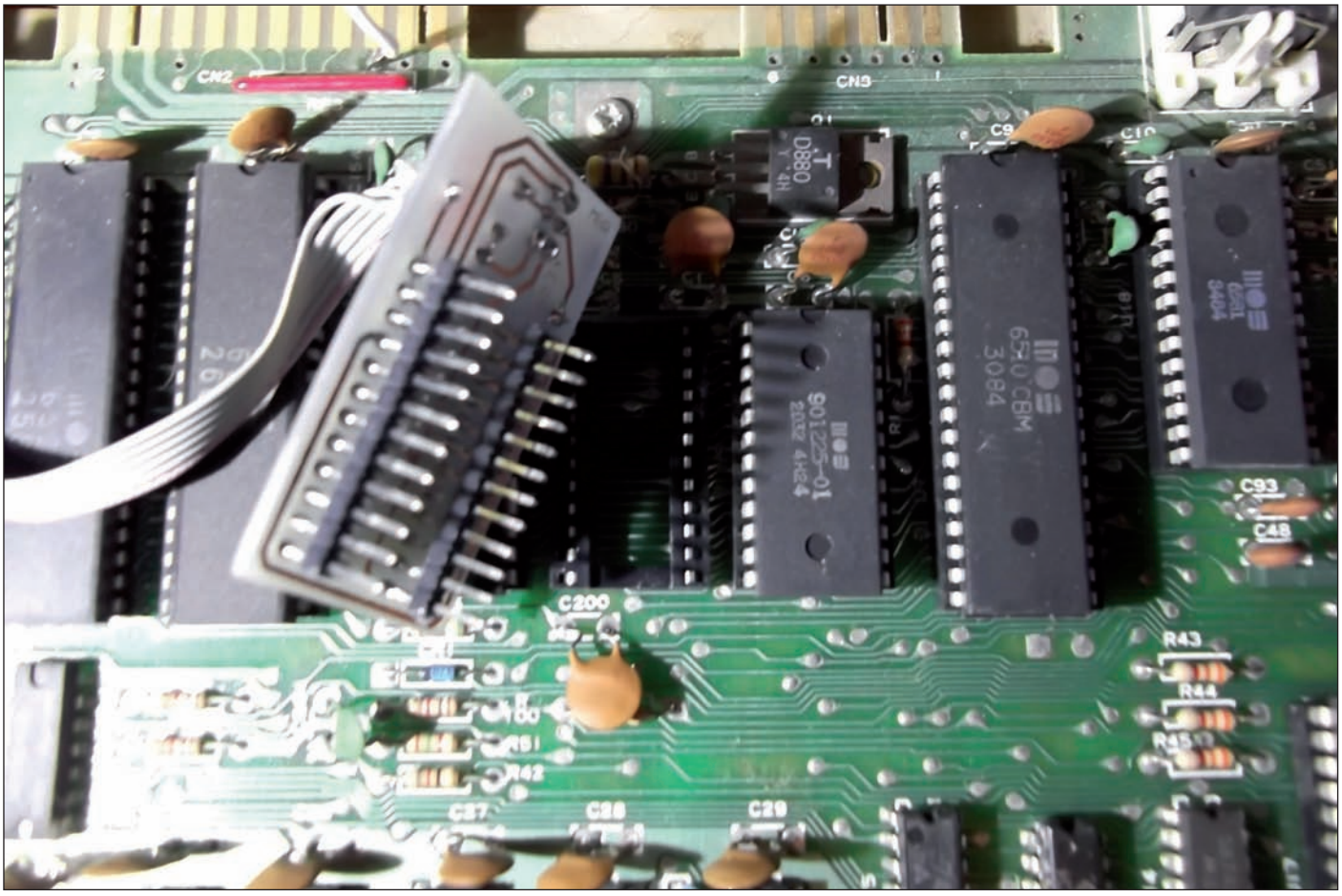



*Figura 6: parte inferiore dell'adattatore*

Dallo schema di **figura 2** notiamo il connettore jp3 denominato bus; qui va direttamente saldato il cavo flat a 6 poli proveniente dal circuito del kernal. Il conteggio binario a 3 bit è affidato a uno dei due contatori contenuti nel cd4520 (dual up-counter) denominato ic3; si tratta di un doppio contatore binario a 4 bit di solo incremento, cioè ad ogni impulso di clock incrementa il valore binario fino ad arrivare a 15 ($f esadecimale) per poi resettarsi e tornare alla posizione zero. Poiché servono solo 3 bit (0-7) per poter selezionare gli otto kernal utilizziamo un apposito pin denominato reset che, se posto a massa (zero logico), non ha alcun effetto mentre, se posto a 1 (+5 volt), esegue il reset riportando il conteggio immediatamente a zero.
Osservando lo schema in **figura 2** tale pin di reset è collegato con il quarto bit Q4-A quindi, appena viene raggiunto il valore di 8, il pin Q4-A si porta al valore logico 1(+5 volt) resettando così il conteggio e riportandosi a zero.
L'impulso di clock è affidato al pulsante SW1 denominato counter; la R1 del valore di 10k mette in pull-down (forzato verso lo stato logico zero) il pin di ingresso clock, R2 e C1, invece, costituiscono un circuito di antirimbalzo, essenziale per evitare falsi conteggi.
Riassumendo: il pin clock, grazie a R1, si trova allo stato logico 0; alla pressione di sw1 giunge una tensione di 5 volt che, attraverso R2, forza lo stato logico del clock a 1; rilasciando SW1 si ritorna allo stato logico 0 generando così l'impulso completo e provocando l'avanzamento del contatore.

La visualizzazione del kernal selezionato è affidata a un display a 7 segmenti con anodo comune pilotato dal decoder-driver CD4543 (IC4) a sua volta comandato dal bus di controllo dell' Eprom.

## MAPPATURA DELLA EEPROM

Come già detto, il multikernal consente di ospitare più banchi da 8 kbyte e poterli scambiare a piacimento. La tabella in **figura 3** mostra come vengono mappati i kernal alternativi all'interno di una sola EPROM da 64 kbyte.
Come si può notare dalla tabella, i kernal alternativi "sostituti" sono allocati al di sopra di 8191: questo perché stiamo utilizzando una EPROM da 64 kbyte anziché da 8 kbyte.
I tre bit restanti (A13), (A14), (A15) determinano il punto di inizio e fine dei vari kernal da 8 kbyte. Proprio questi tre bit sono la chiave del funzionamento del multikernal.

## PROGRAMMAZIONE DELLA EEPROM

Ovviamente per il funzionamento del dispositivo è indispensabile programmare opportunamente la EPROM. I file necessari sono facilmente reperibili in rete tenendo ben presente che devono avere estensione .bin e devono rispettare rigorosamente la dimensione di 8 kbyte (checksum 8192). Per concatenare le otto rom, compresa quella originale, è possibile utilizzare o un semplice editor esadecimale (se ne trovano diversi "freeware" in rete) oppure utilizzare il buon vecchio DOS dal prompt dei comandi di Windows. Per comodità consiglio di rinominare i file relativi alle ROM scaricate con un numero progressivo che rispecchi la vostra scelta; ovviamente andrà sempre lasciata l'estensione .bin e occorrerà conservarli in una cartella che avrete preventivamente creato nella root principale del disco C.
Per esempio: kernal.bin (solitamente quello originale si chiama così) verrà rinominato 1.bin, jiffydos.bin verrà rinominato 2.bin e così via.
Per concatenare i file entrare nel prompt dei comandi di Windows posizionarsi all'interno della cartella contenete i file .bin.
Piccola parentesi per chi non conoscesse il DOS: una volta entrati nel prompt dei comandi digitare cd\, questo comando ci posizionerà nella root principale. Digitare cd "nome della cartella che avete creato"; es. "cd pippo", quindi digitare il seguente comando:
copy /b 1.bin + 2.bin + 3.bin + 4.bin + 5.bin + 6.bin + 7.bin + 8.bin multikernal.bin
verrà creato il file "multikernal.bin". Raccomando di rispettare gli spazi, pena il mancato funzionamento. Inserire il file così creato nella EPROM tramite apposito programmatore.

## MONTAGGIO E COLLAUDO

Presupponendo che non abbiate commesso errori nell' assemblaggio del multikernal, si può si passare al collaudo. Come si vede in **figura 4** l'adattatore EPROM viene installato al posto della ROM originale (MOS901227). Saldare uno spezzone di filo tra JP2 e il pin 3 della user port "Reset". Realizzare un cavetto a 6 poli per connettere il telecomando **(figura 5)** con l'adattatore; si consiglia di far passare il cavo flat attraverso la fessure della user port.
Se tutto è andato a buon fine accendete il C64; sul display dovrebbe apparire la cifra "zero" e il Commodore dovrebbe posizionarsi sul primo kernal scelto (si presuppone l'originale). Alla pressione del tasto counter nel display verrà visualizzato Il numero 1. Premendo il tasto reset, il kernal prescelto sostituirà automaticamente il precedente visualizzandolo sullo schermo.
Buon divertimento. ❑

**CODICE MIP 2827055**

elettroshop.com
brilliant electronics since 1998

Utente anonimo... Login...

Il tuo carrello è vuoto

Home Page | Chi siamo | Listino/Info | Ordina | Download | Contatti | FAQ

CATEGORIE PRODOTTI

- Convertitori Serial-Ethernet
- Convertitori di protocollo
- Bluetooth
- Moduli radio e ZigBee
- Visualizzazione
- Modem GSM/GPRS/GPS
- Interfacce USB e PCI
- CPU e moduli
- Componentistica varia
- Sistemi di sviluppo HW/SW
- Programmatori
- CAE - CAD
- Strumentazione
- CD-ROM e pubblicazioni

RICERCA PRODOTTI

Vai

Ricerca avanzata

NEWSLETTER

Iscriviti alla nostra Newsletter (potrai sempre cancellarti in seguito)

Nome

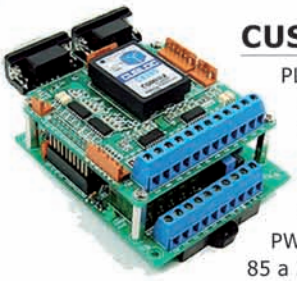
Email

Conferma

## CB210

Il nuovo modulo CB210 è appena disponibile. Combina un controller CUBLOC con un convertitore USV-RS232 su un PCB di facile utilizzo. Basato su ATmega128, ha 80KB di FLASH, 20 linee I/O e convertitore AD.

€ 39,00 Info



## CUSB-14R

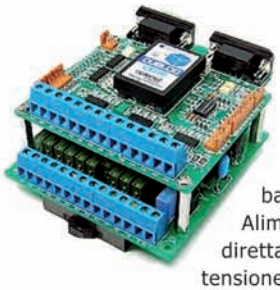
PLC basato su CB280 con 9 ingressi optoisolati e 6 uscite a relè. Ha 6 canali analogici e 6 uscite PWM. Alimentabile da 85 a 264VAC.

€95,00 Info

## CB320

Modulo core Cubloc Basic a 24 pin DIL con controller Ladder Logic. Dotato di 16 linee I/O, 8 ingressi analogici, 3 uscite PWM e 1 timer programmabile. Alimentazione 5/12V.

€ 48,00 Info



## CUSB-22R

Sistema di controllo integrato a 11 ingressi optoisolati e 11 uscite a relè basato su CB280. Alimentabile direttamente con la tensione di rete è dotato di 6 ingressi analogici.

€ 149,00 Info

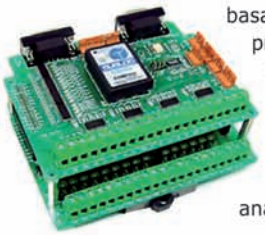
## CB400

Modulo core Cubloc Basic con controller Ladder Logic. Dotato di 83 linee I/O 16 ingressi analogici, 12 uscite PWM e 1 timer programmabile.

€ 59,00 Info

## CUSB-36R

CuSB-36R è un PLC basato su CB280 provvisto di 16 uscite a relè, 16 ingressi digitali, due uscite e due ingressi analogici.



€ 165,00 Info

CODICE MIP 2827191

I prezzi si intendono IVA esclusa

**Micro PLC programmabili in Basic e in Ladder logic**

**COMFILE**
TECHNOLOGY

www.elettroshop.com

# VISUALIZZAZIONE 3D avanzata con PROTEUS

**Come migliorare professionalmente la qualità delle visualizzazioni 3D dei propri progetti, utilizzando contemporaneamente i programmi ARES e Bryce.**

*Figura 1: il logo di ARES.*

## ARES

ARES, strumento di PROTEUS, è un potente programma che mette a disposizione un ambiente completo per l'elettronica. ARES permette di effettuare il posizionamento e lo sbroglio automatico dei circuiti stampati, disegnati con un altro strumento di PROTEUS, ossia ISIS.
ARES include anche un visualizzatore 3D, molto utile e comodo da utilizzare. Esso consente di ottenere un'affidabile anteprima in 3D del circuito appena realizzato, senza costruirlo materialmente. Si tratta indubbiamente di una funzione molto utile. La qualità grafica del 3D ovviamente non è tra le migliori ma ricordiamo che stiamo utilizzando un programma di elettronica, non uno di grafica o di ray-tracing.
Applicando alcune tecniche è possibile realizzare visualizzazioni 3D con un qualità eccezionale, tale a volte da far pensare che il risultato finale sia una foto reale e non un'elaborazione grafica digitale.

## CHE COSA OCCORRE

Per raggiungere l'obiettivo e ottenere visualizzazioni 3D professionali del prodotto finito, occorre utilizzare due programmi distinti:
- Proteus ARES;
- Bryce 7.

Il primo, come anticipato, è il programma che permette di effettuare il piazzamento dei componenti e lo sbroglio automatico dei circuiti stampati.
Il secondo è un potente software per la creazione di grafica 3D e di modellazione solida, con la tecnica del ray-tracing, ottenendo qualità strepitose e immagini reali. Inoltre il programma è fornito gratuitamente dall'azienda che lo produce.

*Figura 2: il logo di Bryce.*

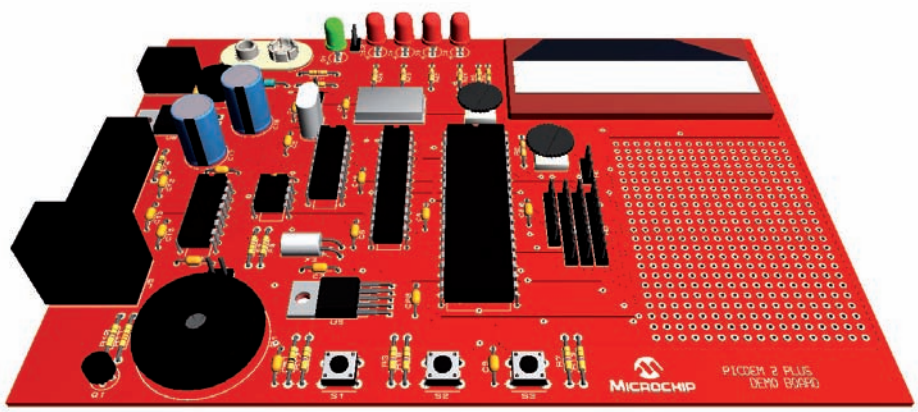



*Figura 3: visualizzazione 3D Standard di ARES.*

di GIOVANNI DI MARIA

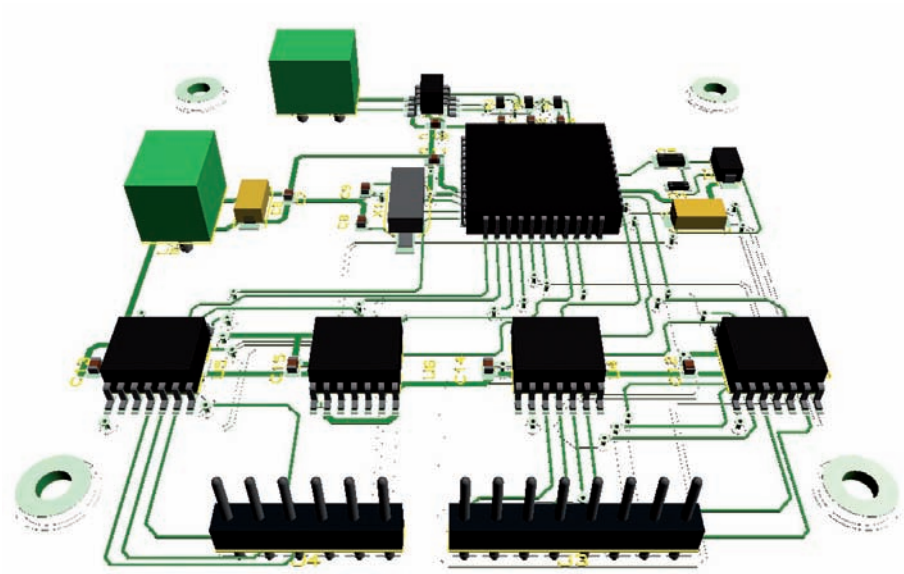


*Figura 4: una basetta in 3D di ARES senza il corretto livello di base.*

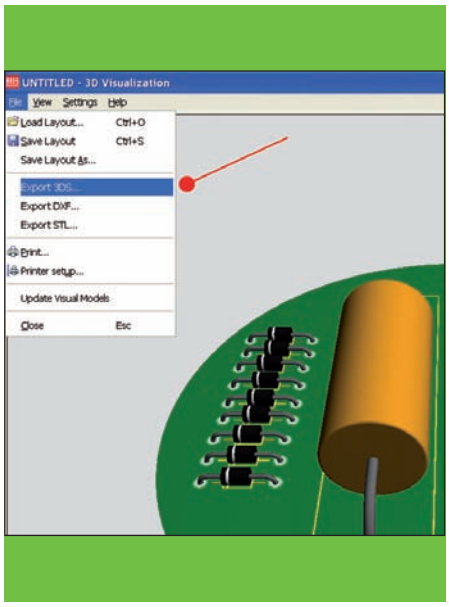



*Figura 6: esportazione del modello 3D in 3DS.*

## VISUALIZZAZIONE 3D CON ARES

Il programma ARES dispone di un'utile funzione 3D che permette la visualizzazione, in anteprima, della scheda elettronica, con tanto di componenti, piste e serigrafie. Si tratta indubbiamente di una caratteristica importante, poiché essa permette anche la visione del prototipo in tutte le angolazioni, con qualsiasi tipo di ingrandimento e di zoom.

Il neo di questa prerogativa è la qualità. Non vengono infatti raggiunti i livelli di accuratezza e veridicità delle immagini che solitamente contraddistinguono le immagini 3D. Non si tratta in ogni caso di una mancanza del programma che, ricordiamolo, è uno strumento di elettronica e non di grafica computerizzata. Con le tecniche spiegate in questo articolo si riuscirà a produrre anteprime tridimensionali straordinarie.

### I fondamenti

Si dà per scontato che il lettore conosca già il programma ARES e il suo utilizzo. Molti aspetti infatti non saranno spiegati in questa sede.
Un circuito stampato, in ARES, può essere creato da zero, direttamente in questo ambiente, oppure è possibile disegnare lo schema nel programma ISIS e importarlo in ARES, con la certezza di avere i componenti collegati correttamente tra

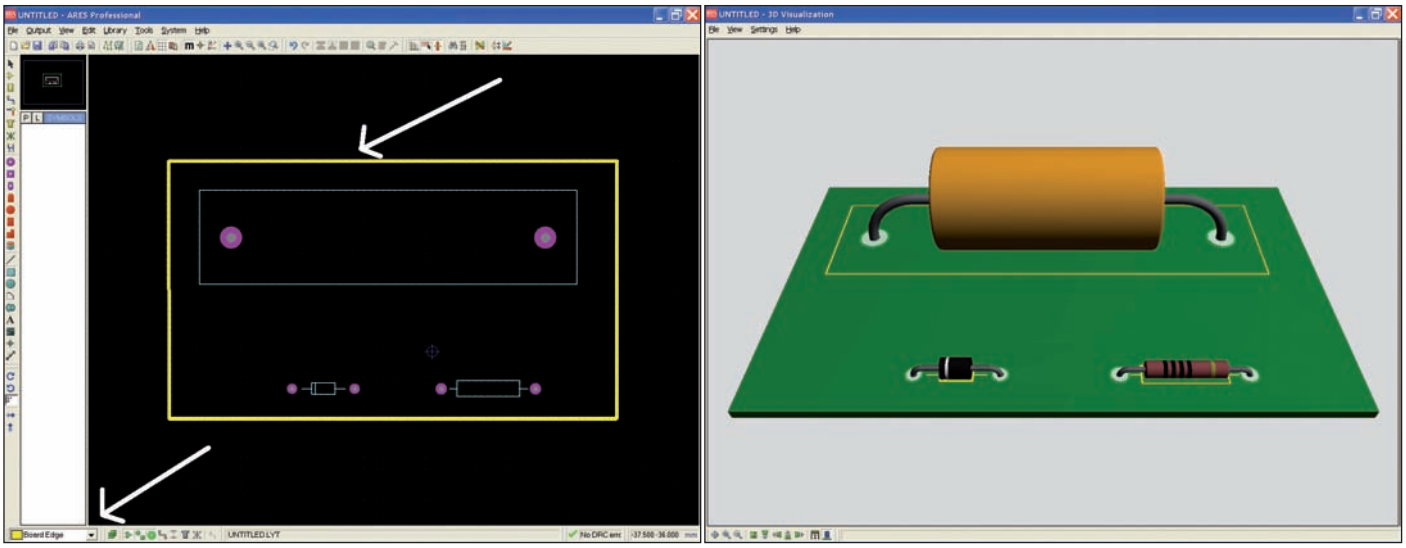



*Figura 5: una basetta con il corretto Board Edge.*

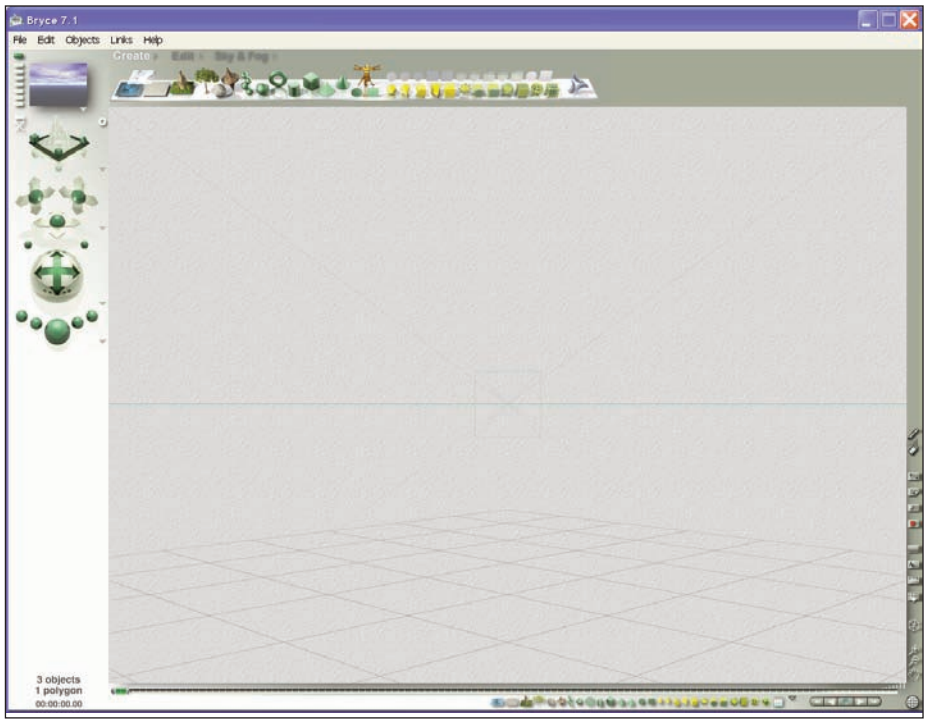



*Figura 7: l'ambiente iniziale di Bryce.*

loro. Il lettore scelga il metodo che più gli è congeniale.
C'è però un passo obbligatorio che deve percorrere al fine di ottenere la corretta visualizzazione della basetta, su cui poggeranno i componenti. Se non si esegue tale procedura, i componenti elettronici sembreranno "volare" in aria, sospinti da qualche misteriosa forza e la basetta risulterà invisibile o, addirittura, inesistente.
La prima cosa da fare è disegnare un rettangolo (o altra forma) attorno ai componenti. Tale bordo conferisce al circuito la quota di ingombro e ne stabilisce, a tutti gli effetti, la reale occupazione spaziale.

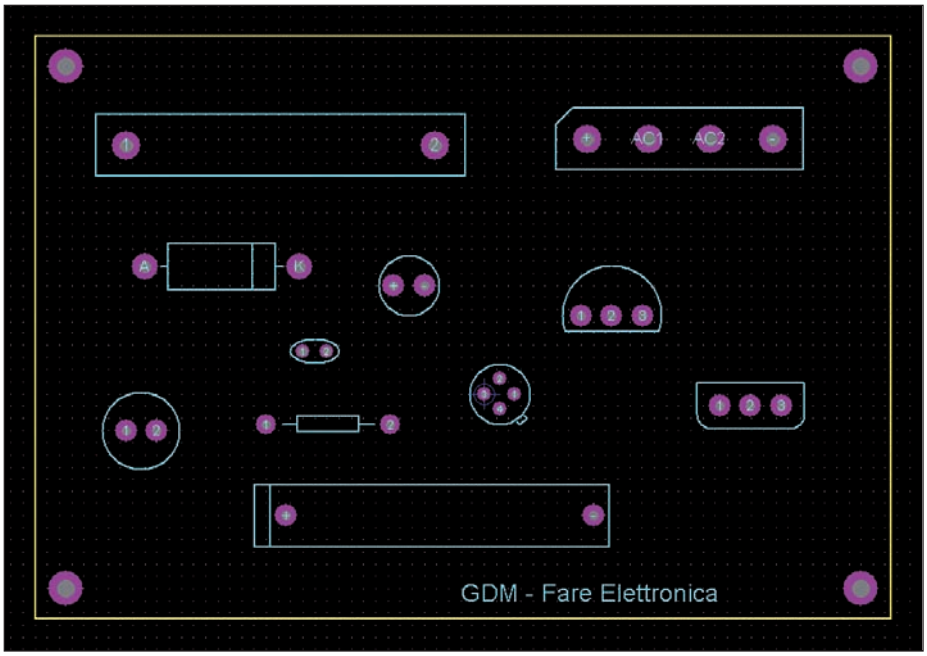



*Figura 8: il circuito in ARES utilizzato per le prove.*

# Next Generation Intelligent LCDs

CODICE MIP 2827195

**RISPARMIA TEMPO.**
**RISPARMIA DENARO.**
**RISPARMIA MANODOPERA.**

**Riduci significativamente i costi di sviluppo**
**Time-to-market ultra-veloce per le tue applicazioni**

Per maggiori informazioni visita il sito **www.elettroshop.com/demmel** oppure chiama il numero **02/66504794**

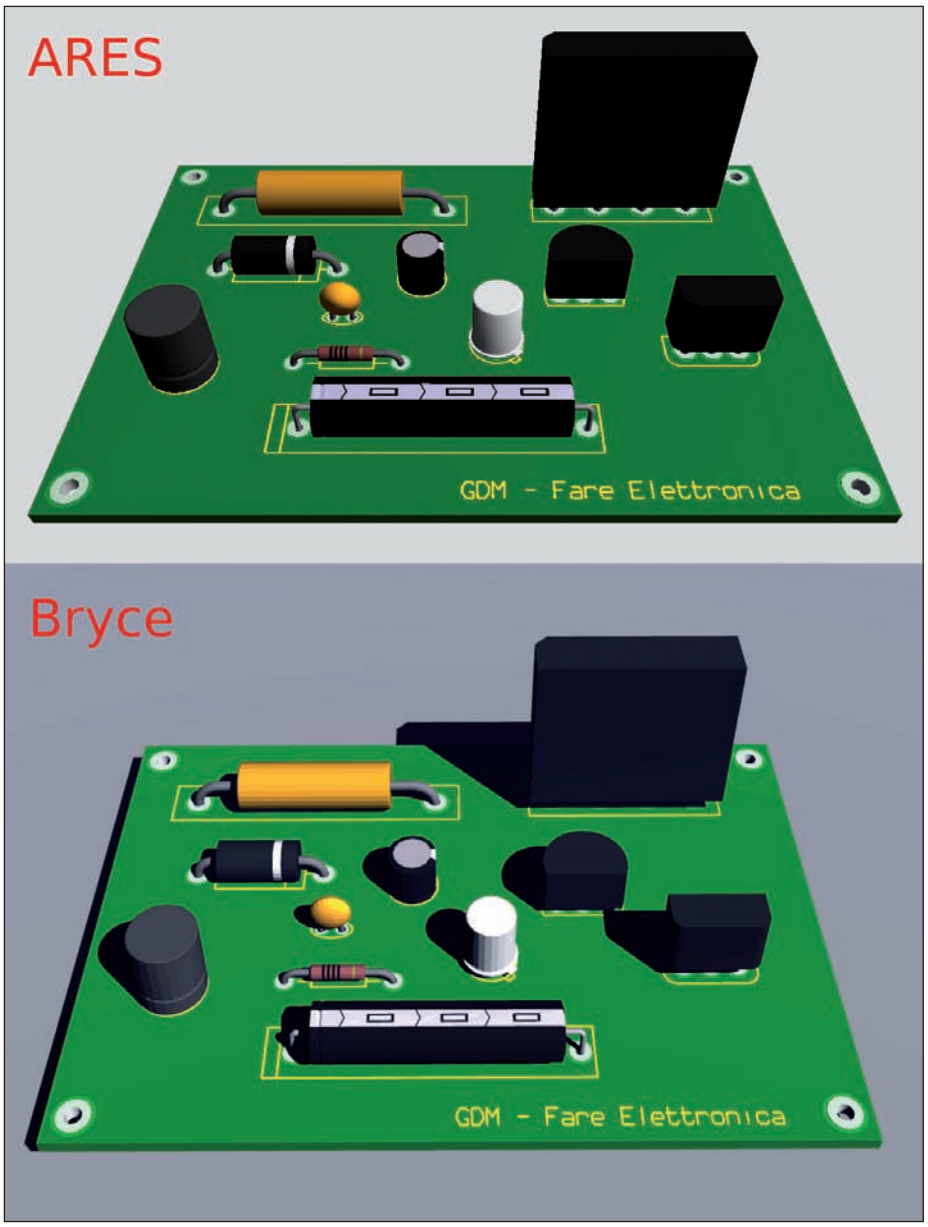



*Figura 9: una prima prova di rendering con Bryce.*

*Figura 10: il pulsante di rendering con Bryce.*

Tale bordo, inoltre deve essere realizzato nel livello "Board Edge", contrassegnato dal colore giallo.
Il Board Edge può essere di qualunque forma, circolare, frastagliata o irregolare: questo permette la realizzazione di un PCB adatto a qualsiasi esigenza.

## COME OTTENERE MAGGIORE QUALITÀ

Le immagini 3D in anteprima possono soddisfare solo a metà il progettista. Certo, vedere un circuito senza averlo costruito è già una grande cosa, ma senz'altro vorrà migliorare la qualità della visualizzazione 3D.
Vediamo quindi, con un po' di lavoro in più, come sia possibile elevare enormemente il livello grafico di rendering, dando quasi la sensazione che il circuito illustrato sia vero e reale.

### Bryce

Per ottenere lo scopo occorre utilizzare un programma di grafica e di modellazione solida: Bryce. Esso è estremamente potente ma semplice al tempo stesso. Non pretendiamo di fornire in questa sede un tutorial sull'utilizzo del software, per il quale il lettore è invitato a scaricare qualche buona guida introduttiva dalla Rete. Seguendo tuttavia le istruzioni dell'articolo, saprà raggiungere un buon grado di visualizzazione 3D, realizzando fedeli rappresentazioni del circuito.

### Esportazione in 3DS

Dopo aver realizzato un circuito e disposto i componenti (con opportuno livello Board Edge) occorre esportare la vista 3D in for-

mato 3DS. Si tratta di un'estensione universale che definisce una tipologia di documento letta dai maggiori programmi di grafica vettoriale. Per salvare il disegno 3D si seguano questi passi:

- in ARES si visualizzi il circuito in 3D, accedendo al menu "3D Visualization", quindi accedendo al menu "File" e alla voce "Export 3DS";
- si scelga quindi una cartella di lavoro nella quale salvare il file 3DS. Questa è tutta la parte di lavoro da svolgere nel programma ARES. Il grosso deve ancora arrivare, quando si utilizzerà Bryce.

## ADESSO LAVORIAMO CON BRYCE

È il momento di effettuare l'importazione del modello 3DS all'interno di Bryce. Si avvii il programma, cliccando l'apposita icona sul desktop. L'ambiente iniziale del programma grafico è rappresentato in **figura 7**.

La prima operazione da effettuare è, ovviamente, l'importazione del file 3DS, precedentemente salvato con ARES. Per importare il modello 3D occorre accedere al menu "File" e alla voce "Import Object", selezionando il file 3DS da aprire. Dopo conferma, il modello apparirà sullo

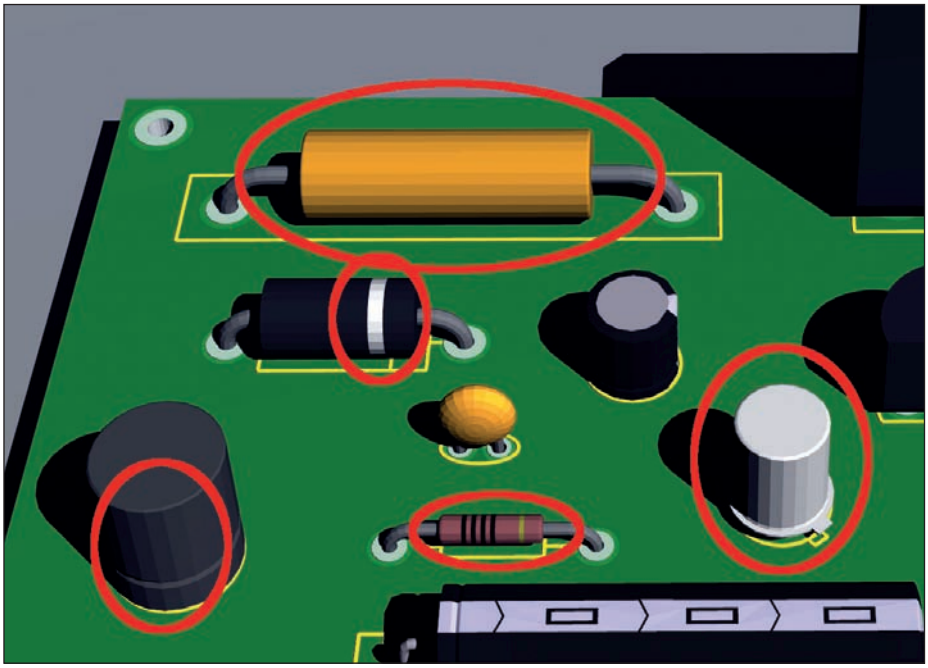


*Figura 11: alla prima importazione manca l'effetto smoothing.*

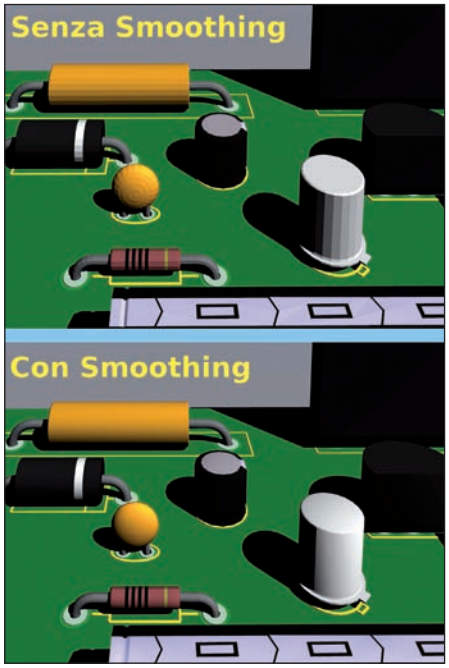



*Figura 13: differenza tra componenti con e senza smoothing.*

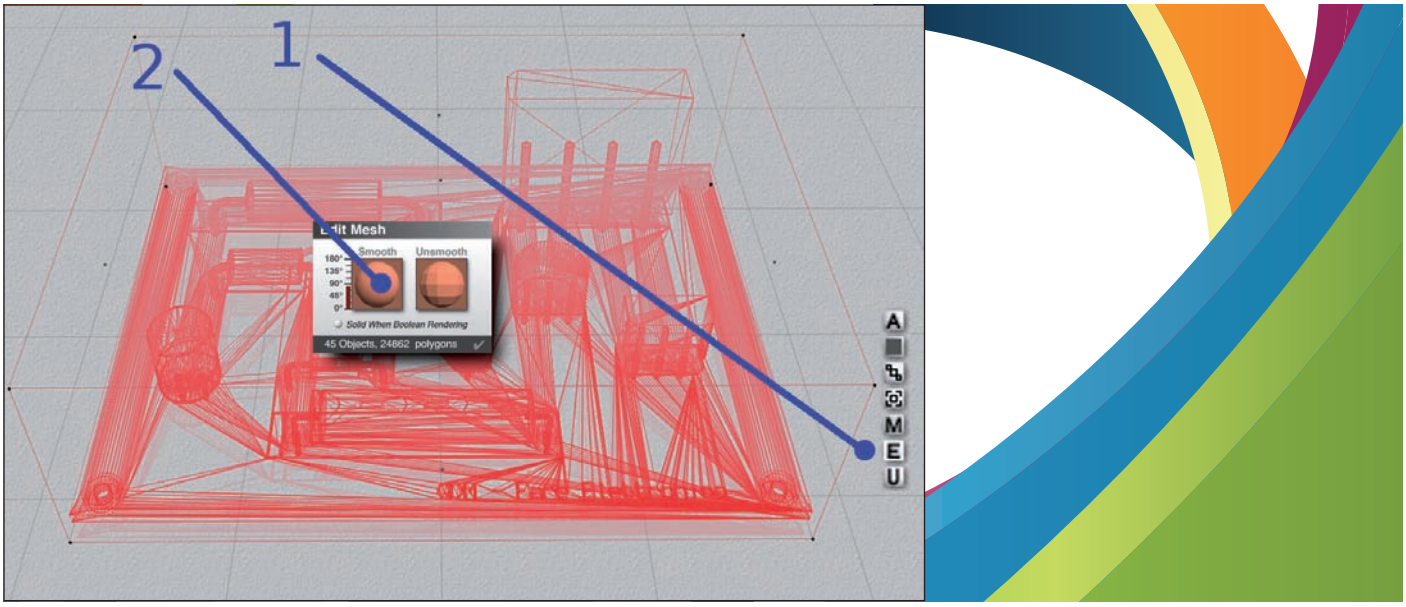



*Figura 12: impostazione dello smoothing.*

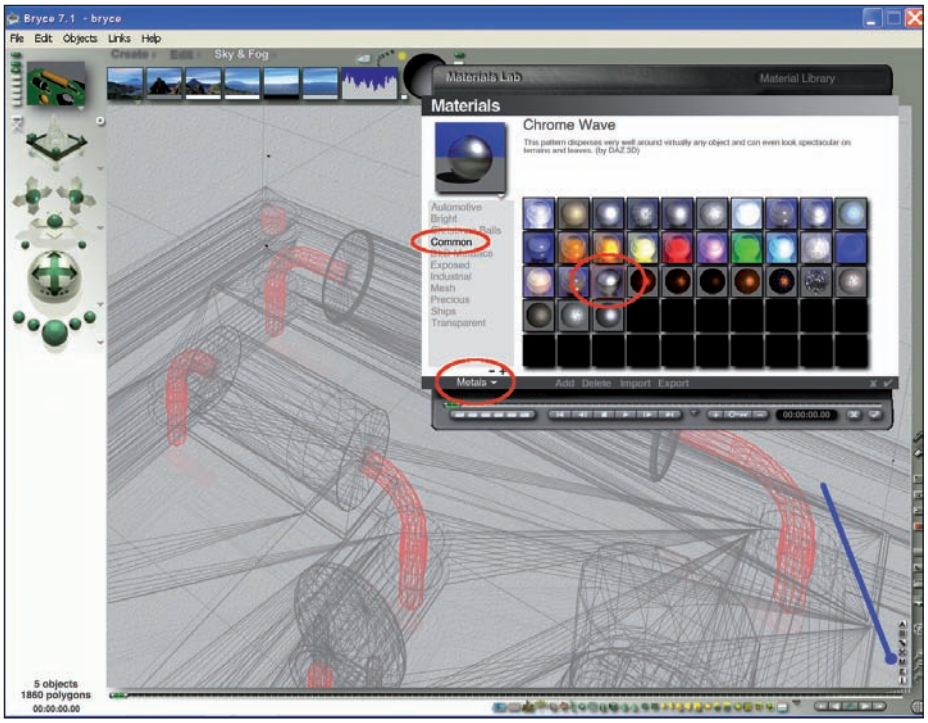



*Figura 14: assegnazione del materiale metallico ai reofori.*

tridimensionale, ossia la proiezione delle ombre portate. Da solo, questo effetto contribuisce a donare quell'effetto di realtà virtuale che, altrimenti, mancherebbe del tutto.
Al primo rendering, ma è del tutto normale, si nota, specialmente nei solidi circolari, uno smusso non perfetto delle curve. È come se le superfici rotonde siano caratterizzate da "scalini". Naturalmente a questo problema esiste una soluzione efficace.
Si deve procedere in questo modo. Occorre selezionare l'intero circuito, con un clic del mouse. Esso diventerà di colore rosso. Si clicchi quindi sull'icona riportante la lettera "E", vicino all'oggetto. Quindi si selezioni, sempre col mouse, la sfera che avvia la procedura di smoothing.
Dopo qualche secondo, avviando nuovamente il rendering, si può notare che tutte le superfici ora sono perfettamente lisce.

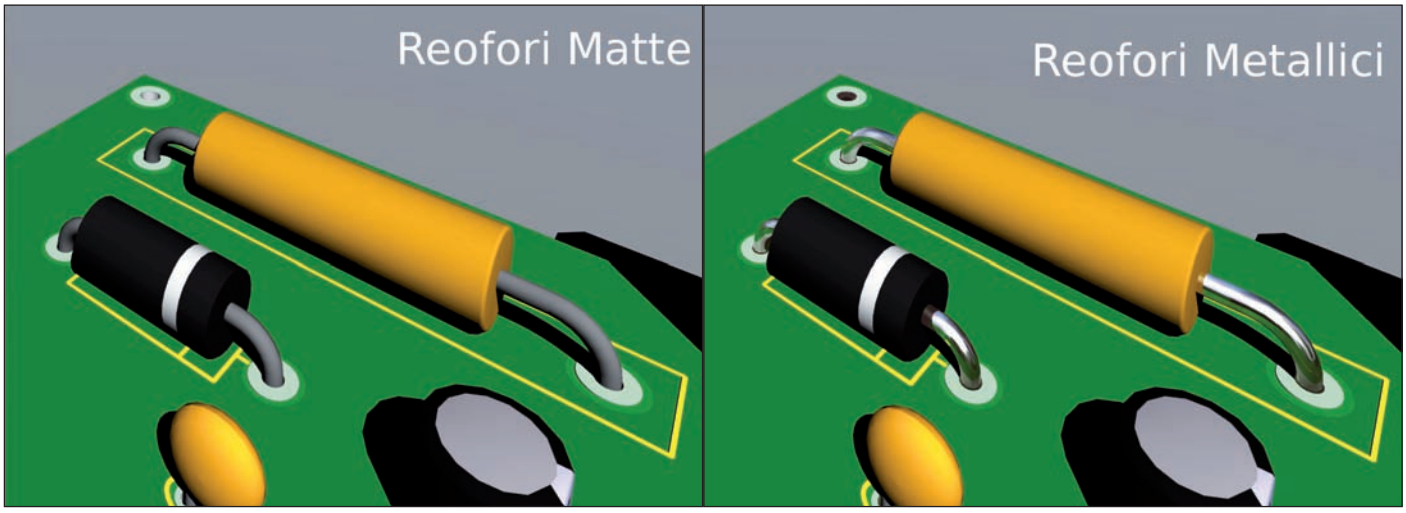



*Figura 15: i magnifici e splendenti reofori metallici.*

schermo in modalità wireframe, ossia formato solo dai segmenti semplici che lo delimitano. Se il disegno è particolarmente complesso e vi sono molte unità primitive, l'operazione potrebbe impiegare anche qualche minuto.

### Primo rendering con Bryce

A questo punto è già possibile abbozzare un primo rendering, di prova, per notare subito le differenze di qualità 3D, rispetto a quello di ARES. Per far questo, si deve premere il pallino verde situato a sinistra, nella barra degli strumenti, proprio sotto la track ball di orientamento. In alternativa si può accede al menu "File" alla voce "Render" (Ctrl + r).
Una prima analisi del risultato decreta la sua indubbia superiorità rispetto al risultato del 3D di ARES, ma ancora molto lavoro deve essere svolto. Una semplice occhiata consente però di rilevare la presenza di un elemento fondamentale, che non deve mai mancare in una immagine

*Figura 16: impostazione della specularità.*

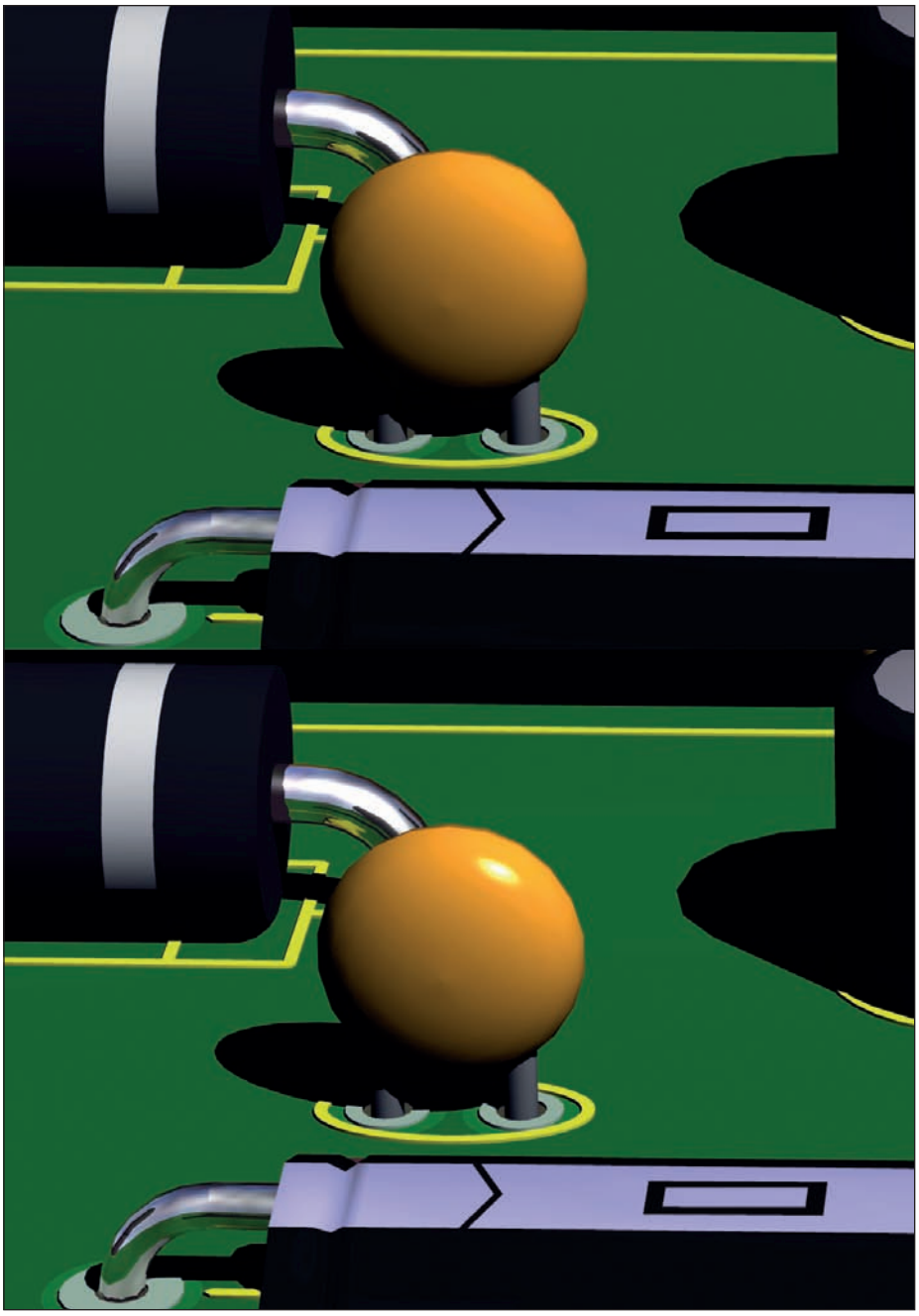


*Figura 17: il realismo aumentato grazie all'alone speculare.*

Alla prima "piccola" delusione iniziale si è sostituita una grande soddisfazione, nel vedere che adesso le superfici sono lisce. La scena 3D deve comunque essere ulteriormente migliorata.

### Reofori di metallo

Un altro tocco di classe che possiamo conferire alla visione 3D del nostro circuito è quello di realizzare i reofori dei componenti di metallo. In questo modo il rendering li rende molto realistici, per la presenza del riflesso tipico che li caratterizza. In più la presenza del "punto" di luce riflessa contribuisce non poco ad aumentare la qualità del risultato finale.
La procedura è la seguente:

- si selezionano i reofori dei componenti desiderati con il mouse e il tasto CTRL;
- si deve premere l'icona con la lettera "M", vicino alla selezione effettuata, per accedere alla libreria dei materiali;

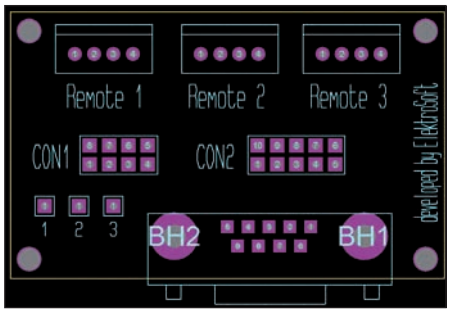



*Figura 18: aggiungere in ARES i testi e fori di fissaggio.*

- si scelga quindi la famiglia dei metalli;
- si scelga poi la sottofamiglia dei metalli comuni;
- si scelga infine il materiale "Chrome Wave" o altro desiderato.

### Alone speculare

Un ulteriore miglioramento è quello di definire un opportuno alone speculare ai materiali opachi. Anche in questo modo il rendering risulta molto più realistico, per la presenza del riflesso di luce. La procedura è la seguente:

- si seleziona il corpo del componente desiderato (o altro oggetto), con il mouse in concomitanza del tasto CTRL;
- si preme l'icona con la lettera "M", vicino alla selezione effettuata, per accedere alla libreria dei materiali;
- si aumenta il valore della specularità.

## ULTERIORI MIGLIORAMENTI

Diamo adesso qualche spunto aggiuntivo per migliorare ulteriormente il grado di realtà virtuale, in aggiunta a quelli che la fantasia del progettista potrebbe regalare.
Una prima miglioria potrebbe essere quella di inserire sulla basetta, in ambiente ARES, alcuni fori in prossimità dei quattro spigoli, per il fissaggio della stessa. Bastano questi pochi elementi per aumentare parecchio la qualità del risultato finale.
Sempre in ambiente ARES, è consigliabile inserire alcuni testi, sia alla base della basetta sia in corrispondenza di componenti di connessione.
Un'altra idea, più complessa, potrebbe essere quella di preparare le mappature di alcune superfici per donare ai componenti una fattezza di stupefacente realtà.

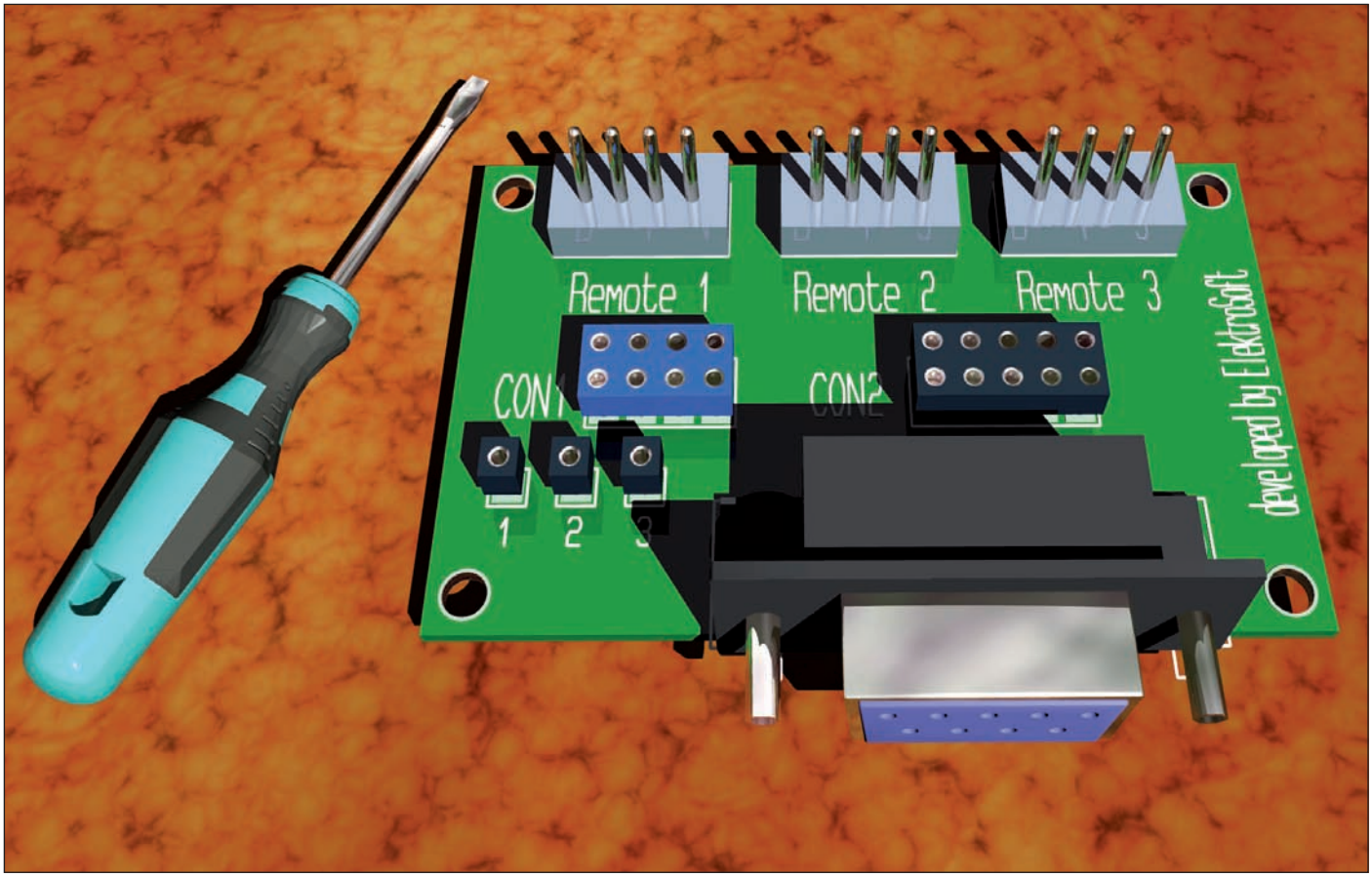



*Figura 19: scena di Bryce con riflessioni, aspetto metallico, testi e altri abbellimenti.*

Ancora, è consigliabile aumentare il grado di riflessione di alcune parti della basetta. Tale operazione si può effettuare sulla superficie, sui condensatori e sui componenti che normalmente sono caratterizzati da un certo grado di riflessione naturale. Anche i fori, i contatti e altre parti di contatto potrebbero essere modificati assegnando loro un materiale metallico. Anche l'aggiunta di ulteriori oggetti alla vista, come attrezzi da laboratorio o altri particolari, possono rendere più realistica la scena finale.

## CONCLUSIONI

Con ciò si conclude questo argomento che, siamo sicuri, susciterà l'interesse e la curiosità di tanti progettisti. I suggerimenti riportati costituiscono alcuni spunti iniziali da seguire, altri ne troverà sicuramente il lettore con la fantasia e l'esperienza. Buon lavoro a tutti. ❑

**CODICE MIP 2827059**

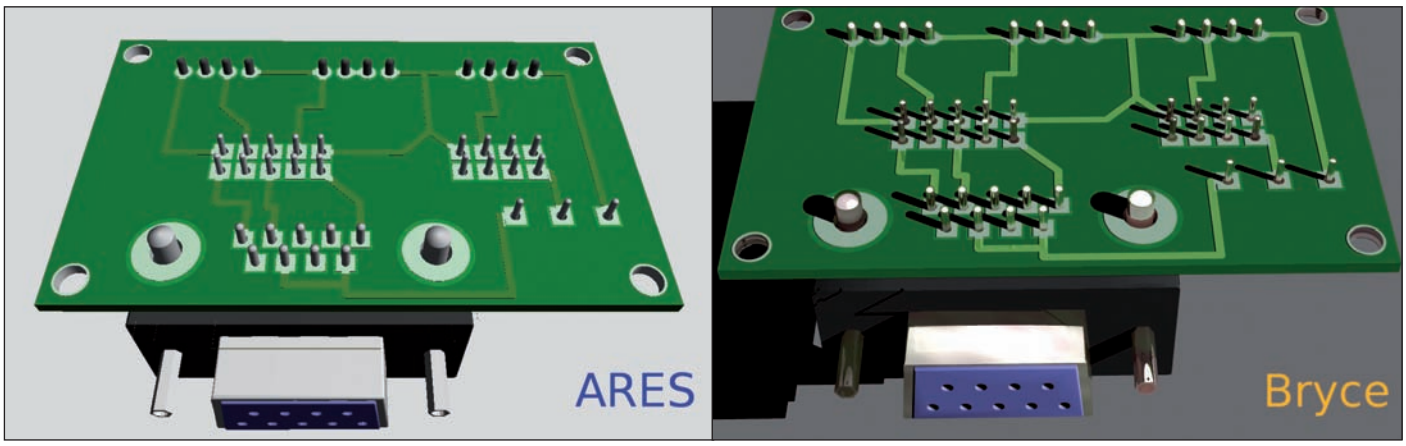


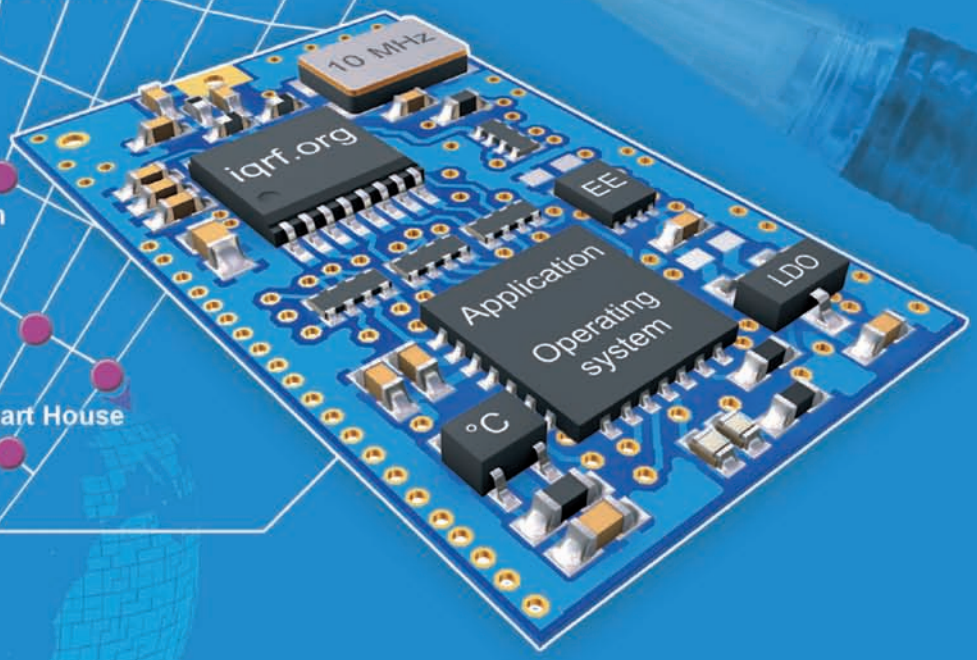

*Figura 20: rendering delle piste del PCB.*

IQRF®

Smarter wireless.
Simply.

Ideale per il collegamento wireless delle lampade a LED!

office automation
telemetria
lighting
industrial control
moduli certificati
easy-to-use
domotica
easy-to-learn
IQRF Smart House
IQMESH




# Reti MESH wireless in pochi passi

sleep 1uA
240 hops
65000 nodi
license free
100 hops in 1 sec
ultra low power
RX 35uA

Moduli radio in 868MHz con PIC a bordo
ICWP - In Circuit Wireless Programming
Fino a 700m per hop, fino a 240 hop
Fino a 65.000 nodi per singola rete
Sistema operativo built-in
Consumi ridottissimi
Programmabili in C
Formato SIM card
Transceiver RF a soli 8 Eur* !

* quotazione per volumi

CODICE MIP 2827199

E' un prodotto distribuito da elettroshop.com
brilliant electronics since 1998

Per maggiori informazioni www.elettroshop.com/iqrf oppure chiamare lo 02/66504794

# CORSO DI PROGRAMMAZIONE *(parte prima)*

# *MikroPascal per PIC*

**Dopo le serie di articoli su MikroBasic e MikroC vi presentiamo, a partire da questa puntata, il MikroPascal, un ambiente di sviluppo per PICmicro anch'esso prodotto da Mikroelektronika. In questa puntata esporremo alcune generalità sui controllori PIC e sul linguaggio O programmazione Pascal, mostreremo come installare il pacchetto MikroPascal e ne analizzeremo l'ambiente di sviluppo (IDE-Integrated Development Environment). Nelle prossime puntate approfondiremo la conoscenza degli strumenti messi a disposizione dall'IDE e inizieremo a usarli compiutamente per la stesura di esempi e progetti firmware.**

## GENERALITÀ

Incoraggiati dal successo riscosso nel recente passato dalle serie di articoli su MikroBasic e MikroC, da questo numero di *Fare Elettronica* cominciamo la pubblicazione di una serie di articoli dedicati al *MikroPascal Pro* per PIC Micro. In questa puntata introduttiva inquadreremo il problema della programmazione dei microcontrollori in maniera da spalancare le porte di questo affascinante mondo a coloro che ancora non hanno esperienza diretta in merito; esamineremo quindi l'ambiente di sviluppo MikroPascal che rappresenterà il nostro principale strumento di lavoro lungo lo sviluppo dell'intero corso. Prima di tutto ciò, al fine di rendere organica l'introduzione al corso e di non lasciare niente al caso illustreremo, seppur brevemente, che cosa si intende per *linguaggio di programmazione*, come nasce la *programmazione strutturata*, che cos'è un *codice assembly*, un codice *sorgente* e un codice *eseguibile*, una *compilazione* e cos'è un *linlking*, come è nato il *Pascal* e quali sono le sue principali caratteristiche.

Dalla prossima puntata cominceremo quindi a proporre una serie di esempi applicativi di firmware per PIC Micro sviluppati con *MikroPascal*. Al momento opportuno faremo riferimento ad aspetti che riguardano l'implementazione hardware delle applicazioni e l'architettura interna dei controllori che prenderemo a riferimento per l'implementazione del firmware. Al fine di dare l'opportunità al lettore di sperimentare quanto verrà presentato, non mancheremo di fare riferimenti a *demo-board*, *schede di sviluppo*, *programmatori* e persino a strumenti di simulazione software di microcontrollori PIC. Il percorso che seguiremo tenderà, in sintesi, a privilegiare aspetti pratici di base senza tuttavia trascurare nozioni tecniche teoriche utili alla comprensione degli argomenti anche da parte di chi si avvicina per la prima volta al mondo dei microcontrollori.

Detto questo, riteniamo opportuno definire in maniera chiara tre aspetti:

1) qual è lo scopo del corso;
2) a chi è rivolto;
3) in che modo, secondo quale percorso e con quali mezzi si intende perseguire lo scopo.

Chiunque voglia oggi imparare a programmare i controllori PIC è alla ricerca, quasi sempre, di un percorso didattico che si riveli allo stesso tempo *semplice* ed *efficace*. Semplice significa anche, in una certa misura, intuitivo; efficace significa invece anche, in una certa misura, diretto al problema. Per perseguire una finalità di questo tipo è necessario per prima cosa dotarsi di strumenti adeguati, quindi imparare ad utilizzare questi ultimi nella maniera più razionale e proficua possibile, applicandoli alla risoluzione di problemi specifici.

Un progetto che impieghi buoni strumenti, utilizzati correttamente mostra, quasi sempre, o con più probabilità, uno sviluppo lineare, scalabile e modificabile con facilità anche da chi non lo ha concepito o sviluppato dall'origine. Cercheremo per questo, da un lato di avere un approccio immediato agli argomenti, dall'altro di non trascurare una corretta impostazione delle problematiche e della loro risoluzione, al fine di non rendere gli esempi pratici, per quanto semplici, fi-

di ANTONIO GIANNICO

ne a sé stessi. Lo scopo del corso non sarà unicamente quello di insegnarvi a utilizzare il MikroPascal ma, più in generale, quello di introdurrvi, specie coloro che sono alle prime armi, nel mondo dei microcontrollori e della programmazione dei microcontrollori attraverso l'impiego di linguaggi di programmazione ad alto livello.
Uno dei motivi, ma non il solo, per il quale abbiamo scelto di proporre questo corso sul compilatore MikroPascal, per PICmicro è che è possibile scaricarlo gratuitamente dal sito di *Mikroelektronika*, sebbene, occorre puntualizzarlo fin da subito, in versione *demo* limitata nella dimensione massima del programma compilato.
L'elettronica si è ormai compiutamente evoluta, adeguandosi e integrandosi sempre più con quella che un tempo, detto in maniera relativamente semplicistica e approssimata, era percepita come informatica. Il risultato di questa evoluzione è che le applicazioni elettroniche di oggi non sono quasi mai puramente analogiche o puramente digitali e nemmeno mixed-signal, ma piuttosto programmabili. Qualunque apparecchiatura elettronica da noi utilizzata comunemente (telefoni cellulari, televisori, videogiochi, centraline di allarme, elettronica a bordo veicoli ecc.) contiene dispositivi programmabili. Diversamente da quanto accadeva agli albori dell'elettronica programmabile, l'elettronica embedded è oggi programmata utilizzando veri e propri linguaggi di programmazione ad alto livello, del tutto simili a quelli utilizzati per la realizzazione di applicazioni software. Il *progettista elettronico* è per questo sempre più spesso chiamato ad aggiungere alle competenze tradizionali di *analista e progettista di circuiti* quella di *programmatore firmware*. L'elettronica, diventando programmabile, ha compiuto un'evoluzione epocale in termini di funzionalità e di flessibilità delle applicazioni.
Il corso è in ultima analisi rivolto sia a coloro che pur non avendo alcuna esperienza in merito desiderano avvicinarsi alla programmazione dei sistemi embedded e dei microcontrollori, sia a coloro che pur avendo un certa esperienza in merito non conoscono o non hanno ancora avuto modo di utilizzare MikroPascal. Chi già ha avuto modo di utilizzare MikroC e/o MikroBasic per PIC, che sono prodotti dalla stessa casa, noterà numerose somiglianze fra i tre pacchetti, se non nel modo in cui viene scritto il codice, almeno nel modo in cui è organizzato l'IDE e nelle utility che lo stesso integra.

Fondamentalmente, per un corretto raggiungimento dei nostri obiettivi, sarà necessario:
- l'analisi della sintassi di base del linguaggio e dei costrutti;
- l'impiego efficace delle utility che l'ambiente di sviluppo integra;
- l'analisi e l'impiego delle numerose funzioni di libreria che l'ambiente di sviluppo integra;
- fare di tanto in tanto cenni ad accorgimenti opportuni nella stesura del codice, all'impiego corretto dei tipi di dati e all'ottimizzazione del codice sia da un punto di vista di leggibilità che della compattezza.

Molti degli esempi pratici di progetto che saranno mostrati nel corso del cammino potranno essere rivisitati, modificati o ampliati dal lettore oppure essere semplicemente di spunto per nuovi progetti.

## UNA NOTA IMPORTANTE

Questo articolo e quelli che seguiranno non intendono in nessun modo sostituire i manuali d'uso ufficiali sul MikroPascal resi disponibili da *Mikroelektronica*. Di conseguenza, ogni approfondimento sull'impiego del compilatore MikroPascal e dell'ambiente di sviluppo integrato va fatto sulla documentazione ufficiale di Mikroelektronica, peraltro direttamente scaricabile dal relativo sito (www.mikroe.com) oppure consultabile attraverso l'Help in linea dell'ambiente integrato.

## CHE COS'È UN MICROCONTROLLORE

Se utilizziamo un *compilatore per microcontrollori* è certamente perché vogliamo sviluppare un'applicazione firmware che dovrà girare su un microcontrollore in maniera tale da gestire (controllare, appunto) un determinato *processo* secondo un determinato *algoritmo*. Appare ovvio pertanto la necessità di avere un'idea abbastanza chiara di che cosa sia un microcontrollore.
Fisicamente un microcontrollore è un circuito integrato ( micro + controllore cioè controllore integrato in un unico microchip) che si differenzia da un comune circuito integrato per il fatto che la funzione da esso svolta non è impressa a priori nel suo hardware ma è imposta dal programmatore attraverso il programma (firmware) che egli carica nella sua memoria flash. Tale programma rappresenta l'implementazione di un algoritmo dedicato alla gestione di una specifica *funzione, applicazione o processo*. Il tipo di processo controllato e la sua complessità dipendono dalle risorse fisiche che il controllore integra e in qualche misura dall'abilità del programmatore e dalla flessibilità degli strumenti che egli ha a disposizione per sviluppare il firmware.
Senza entrare troppo nel dettaglio è possibile, fin da ora, comprendere come un microcontrollore debba necessariamente integrare le seguenti risorse fisiche:
- una *CPU* (Central Processing Unit) che esegue in successione le *istruzioni di codice eseguibile* ottenute dalla compilazione di un *codice sorgente*;
- una *PROM* (Progammable Read Only Memory) necessaria a contenere il programma (firmware) compilato che implementa, sotto forma di codice eseguibile, l'algoritmo specifico;
- una *EEPROM* (Electrical Erasable Programmable Read Only Memory) in grado di mantenere immagazzinati i dati anche in assenza di alimentazione;
- una memoria *RAM* (Random Access Memory) contenente provvisoriamente dati transitori, cioè i valori assunti da quelle che comunemente chiamiamo variabili;
- PORTE di *I/O* attraverso le quali il controllore si interfaccia con il mondo esterno per acquisire dati in ingresso e per fornire risultati, comandi di attuazione ed il controllo di specifici circuiti in uscita.

Molti, a causa della famigliarità che un po' tutti abbiamo con il mondo dei computer sono portati ad assimilare il *microcontrollore* al *microprocessore*. In effetti,

entrambi *eseguono istruzioni ed elaborano informazioni secondo algoritmi definiti dall'utente* ed espressi mediante un codice di programma. Tuttavia, un microcontrollore presenta differenze sostanziali rispetto a un microprocessore, tra cui:

- memoria flash e RAM direttamente integrata all'interno dello stesso chip;
- *I/O* analogici e digitali integrati;
- periferiche di comunicazione quali *UART, USB, Ethernet, I2C, SPI, CAN* spesso integrate anch'esse.

L'altra differenza che ai più appare evidente è costituita dalla ridotta potenza di calcolo e dalla ridotta disponibilità di risorse fisiche rispetto a quelle proprie dei microprocessori (memoria in primo luogo), caratteristiche che rendono i microcontrollori idonei alla realizzazione di specifiche applicazioni dedicate.

Prima di proseguire oltre, alcune puntualizzazioni doverose. Vi siete mai chiesti che cosa significhi PIC? Ebbene, si tratta di un acronimo che sta per *Peripheral Interface Controller*. Si tratta di microcontrollori basati sulla tipica *architettura Harvard* che spaziano dalla famiglia a 8 bit alle famiglie rispettivamente a 16 e 32 bit fino alla famiglie dei cosiddetti dsPIC che presentano core integrante un DSP. Si spazia in questo modo dai controllori PIC con più ridotte risorse e potenza di elaborazione a quelli con maggiori risorse e maggiore potenza di elaborazione, cosa che in una certa misura si riflette anche nell'aspetto hardware che essi mostrano all'esterno anche a prima vista poiché si passa dai più piccoli in package DIP a soli 6 pin ai più grandi in package SMT a 144 pin.

Per il momento non ci dilunghiamo oltre sull'architettura e su aspetti hardware dei controllori PIC ma ci dedichiamo piuttosto all'ambiente di sviluppo MikroPascal. Il fatto che il corso sia dedicato alla programmazione in MikroPascal dei controllori PIC di Microchip trae origine dalle seguenti considerazioni:

- i PIC Micro sono ampiamente diffusi;
- presentano costi contenuti e sono facilmente reperibili;
- presentano sul mercato diversi sistemi di sviluppo software e hardware (ambienti di sviluppo, e demo-board) dal costo relativamente contenuto ed ampiamente documentati;
- sono presenti sul mercato diversi compilatori basati su diversi linguaggi ad alto livello alcuni dei quali liberamente scaricabili e utilizzabili in versione demo (per es. MikroC, MikrBasic e MikroPascal);

## LINGUAGGI DI PROGRAMMAZIONE

In informatica, un *linguaggio di programmazione* è un linguaggio formale, dotato (al pari di un linguaggio naturale) di un lessico, di una sintassi e di una semantica ben definite, utilizzabile per la scrittura di un programma che definisca il comportamento di una applicazione software destinata a "girare", generalmente, su un computer. L'assembly è il primo vero linguaggio di programmazione che ha consentito, nell'evoluzione della scienza dei calcolatori, il distacco del programmatore dal cosiddetto *linguaggio macchina*. Esso infatti costituisce una rappresentazione simbolica del linguaggio macchina e per questo appare inevitabilmente spostato verso il linguaggio macchina stesso cioè verso l'hardware e la sua architettura. In sintesi, si usa dire che l'assembly è un *linguaggio di basso livello*. L'aspetto peculiare di un tale linguaggio è che la sola forma di controllo di flusso è *l'istruzione di salto condizionato*.

Questo aspetto è centrale nella comprensione delle differenze che sussistono tra linguaggio a basso livello e linguaggio ad alto livello ed è ciò che porta a scrivere programmi difficili da seguire da un punto di vista del flusso proprio a causa dei continui salti da un punto all'altro del codice. Per questo motivo, i linguaggi di programmazione ad alto livello sono nati con il fine di concretizzare l'allontanamento da un approccio al controllo del flusso così minimale, introducendo strutture di controllo (e anche strutture dati) più evolute. Questo sforzo portò inizialmente alla nascita del linguaggio Fortran, nel 1957 ad opera di John Backus, da cui derivò successivamente il BASIC (1964). Nel 1970 Niklaus Wirth pubblicava il Pascal, il primo vero linguaggio strutturato a scopo didattico; negli anni successivi fu introdotto con grande successo presso i programmatori il C. Oltre al salto condizionato (istruzione IF), questa nuova generazione di linguaggi introduceva nuove strutture di controllo di flusso come i cicli WHILE e FOR e le istruzioni CASE e SWITCH. Le conseguenze più evidenti che si ebbero nella programmazione furono:

- la naturale diminuzione del ricorso alle istruzioni di salto (GOTO);
- la possibilità (conseguente al precedente punto) di scrivere, con una certa naturalezza, codice più chiaro ed elegante, più facilmente decifrabile e quindi anche di più facile sviluppo e manutenzione.

Tutti i linguaggi di programmazione esistenti continuavano in ogni caso a basarsi due concetti chiave: quello di variabile (un dato o un insieme di dati cui corrisponde un certo numero fisso o variabile di locazioni di memoria che vengono allocate, cioè riservate, per contenere i dati stessi) e quello di istruzione (in sintesi un comando che agendo su determinati dati determina il cambiamento dello stato di un processo e quindi in pratica dello stato interno del calcolatore).

## CARATTERISTICHE INTRINSECHE DI UN LINGUAGGIO DI PROGRAMMAZIONE

In generale, le caratteristiche del linguaggio condizionano direttamente lo stile di programmazione del programmatore. Il BASIC, per esempio, è un linguaggio facile da imparare, grazie alle sue caratteristiche peculiari: poche regole e una sintassi chiara e semplice. Il Pascal non solo presenta questi pregi tipici del BASIC ma è anche un linguaggio didattico poiché "educa" il neo-programmatore a uno stile di programmazione che aiuta, per sua natura, a evitare errori. Al contrario, il C non è un linguaggio didattico perché ha una semantica più complessa che lo rende più efficiente e flessibile ma che richiede al tempo stesso maggiore esperienza.

La leggibilità di un codice sorgente può essere certamente facilitata dalla specifica sintassi che il linguaggio prevede ma in ogni caso dipende anche dallo stile di programmazione del programmatore. Una caratteristica per molti aspetti preferibile in un linguaggio è comunque la *robustezza* ovvero la caratteristica di essere strutturato in maniera tale da prevenire implicitamente gli errori di programmazione. Questa caratteristica si solito si ottiene in un linguaggio imponendo un controllo

piuttosto stretto sui tipi di dati e una sintassi abbastanza rigida. Il Pascal è il tipico esempio di linguaggio robusto e non potrebbe essere altrimenti, essendo nato per scopi didattici; una irregolarità, nell'uso dei tipi di dati in particolare, viene immediatamente assunta dal compilatore come frutto di un errore del programmatore. È esattamente l'opposto di quanto avviene con l'assemby in cui non può esserci, per sua natura, una forma di controllo così puntuale.

## PROGRAMMAZIONE STRUTTURATA

Le idee chiave della programmazione strutturata discendono da considerazioni riguardanti la struttura di controllo del salto incondizionato (o goto, "vai a"). In un celebre articolo del 1968, "*Goto statement considered harmful*", Edsger Dijkstra discuteva gli effetti negativi del goto sulla leggibilità e modificabilità del codice che assumeva il cosiddetto aspetto di "codice a spaghetti". Una celebre pubblicazione, fondamentale nell'affermazione della programmazione strutturata "*Flow Diagrams, Turing Machines, and Languages with Only Two Formation Rules*", di Corrado Böhm e Giuseppe Jacopini dimostrava inoltre che qualsiasi programma scritto usando il goto poteva essere riscritto senza, a patto di avere a disposizione tre tipi di strutture di controllo: *sequenza, ripetizione e alternativa*. Nasceva in questo modo la programmazione strutturata che aveva come scopo quella di rendere migliore la struttura e il controllo del codice, vincolando il programmatore a evitare le istruzioni di salto incondizionato e a usare le sole strutture di controllo canoniche di sequenza, selezione (if…then…else, if…then) e ciclo (while…do…, repeat…until…). Il Pascal è per questo, come il C, un linguaggio strutturato. Ciascuna struttura di controllo tende infatti, in maniera naturale, a definire un àmbito isolato di azione che facilita la progettazione top-down delle applicazioni e lo sviluppo per affinamenti successivi. I linguaggi di programmazione strutturati iniziarono a emergere intorno all'inizio degli anni settanta. La possibilità di uso del goto fu generalmente mantenuta ma divenne fortemente sconsigliata.

## CODICE SORGENTE, COMPILAZIONE, INTERPRETAZIONE, LINKING

Programmare in un dato linguaggio di programmazione significa molto semplicemente scrivere uno o più file di testo (quindi ASCII) che nel loro insieme costituiscono il cosiddetto codice sorgente. I font e l'aspetto grafico di questi sono sostanzialmente irrilevanti ai fini dell'algoritmo espresso. Ne consegue che a un programmatore, di fatto, non servono programmi di videoscrittura per scrivere il codice; in linea di massima è sufficiente un editor di testo. In realtà, oggi tutti gli editor di testo utilizzati dai programmatori risultano integrati con altri strumenti di lavoro. Si tratta di *compilatori, linker, interpreti* e *utility* di ogni genere che facilitano la stesura, l'analisi e il debug del codice. In questo caso non si parla più semplicisticamente di editor ma piuttosto di *IDE* (*Integrated Development Environment*) o ambiente di sviluppo integrato.
Il codice sorgente può essere eseguito passandolo a un interprete che esegue le istruzioni in esso contenute (è ciò che accade tipicamente per i linguaggi di scripting) oppure può essere compilato, cioè tradotto in istruzioni di linguaggio macchina da un programma detto *compilatore* che genera un file binario o cosiddetto "eseguibile" (codice eseguibile) poiché non ha bisogno di altri programmi per essere eseguito (da cui la maggiore velocità rispetto a un programma interpretato). La compilazione è quindi il processo grazie al quale, il programma sorgente, scritto in un linguaggio di programmazione ad alto livello, viene tradotto in codice eseguibile per mezzo di un altro programma detto compilatore.
Se il programma, come spesso accade, usa librerie o è composto da più moduli software, questi devono essere "collegati" tra loro. Questa operazione è eseguita attraverso uno strumento detto linker ("collegatore"). Un approccio di questo tipo incoraggia la modularità implementativa del programma e quindi ne facilità indirettamente la leggibilità, lo sviluppo e la manutenzione poiché consente di isolare più facilmente parti di programma contenenti bachi e di correggerle o modificarle mantenendole ben distinte.

Il vantaggio della compilazione rispetto all'interpretazione sta nell'efficienza nettamente superiore in termini di prestazioni; lo svantaggio sta invece nel fatto che è un processo vincolato alla piattaforma cioè alla combinazione di architettura hardware e sistema operativo nel caso di un PC, all'architettura del processore o del controllore nel caso di applicazioni embedded. Questo appare evidente, per esempio, osservando che esistono compilatori MikroC, MikroBasic e MikroPascal distiniti per ciascuna delle seguenti piattaforme hardware: PIC, AVR, dSPic30/33 e PIC24, PIC32, 8051, ARM. A tale proposito è possibile trovare conferma direttamente sul sito di Mikroelektronika (www.mikroe.com), l'azienda che produce i suddetti compilatori.

## IL LINGUAGGIO PASCAL

Il Pascal è un linguaggio di programmazione nato a opera di Niklaus Wirth nel 1970, che si presta a scopi didattici e che risulta caratterizzato da una forte tipizzazione statica. Influenzato dal linguaggio ALGOL, fu così chiamato in onore del matematico e filosofo francese Blaise Pascal, inventore della prima macchina calcolatrice automatica. Wirth, docente di programmazione, comprese la mancanza di un linguaggio di programmazione adatto alla didattica. All'epoca era già abbastanza diffuso il BASIC che, creato nel 1964, appariva semplice ma non dotato di strutture dati avanzate e non strutturato. Queste considerazioni portarono Wirth a creare il Pascal, enfatizzando proprio il concetto di programmazione strutturata.
Egli riuscì a proporre un linguaggio pensato per utenti inesperti, poiché i compilatori Pascal si comportano in maniera "materna", nei confronti del programmatore, partendo dal presupposto che qualunque irregolarità del codice o presunta tale, specie nell'impiego dei tipi di dati, rappresenti di fatto un errore. Esso sottopone in tal modo il programmatore a una precisa disciplina che lo educa e lo spinge in maniera naturale ad adottare uno stile di programmazione razionale e ben strutturato.
Il confronto tra un programma scritto in Pascal e lo stesso scritto, per esempio in

Basic, mostra mediamente una maggiore chiarezza e linearità nel primo piuttosto che nel secondo.
A conferma di quanto Wirth fosse riuscito nei propri intenti, il Pascal si diffuse ben presto negli ambienti scolastici. La sua evoluzione ha portato alla realizzazione di versioni a oggetti del Pascal, come il Turbo Pascal e l'Object Pascal successivamente utilizzato nell'ambiente di Delphi Borland.
In concreto e in sintesi, le caratteristiche del Pascal possono essere riassunte nei seguenti punti:
- sintassi estremamente chiara sebbene relativamente rigida;
- obbligo di dividere il programma in sezioni ben definite;
- obbligo di dichiarare in anticipo tutte le variabili usate nel programma;
- uso di tipi di dati complessi e definibili dal programmatore, detti *record*;
- uso dei puntatori e allocazione dinamica della memoria.

Il Pascal è, per questo motivo, classificato come linguaggio semidinamico (consente di gestire i puntatori, permettendo di creare liste di variabili di dimensione non nota all'atto della compilazione). Al posto delle subroutine, di uso comune nei linguaggi di allora, il Pascal introduceva le procedure e le funzioni che incapsulano al loro interno variabili e dati locali esponendo all'esterno i soli valori di ingresso e di uscita. Nella sua formulazione originale, proprio per la filosofia che ne aveva ispirata la nascita, era privo dell'istruzione GOTO, aggiunta solo in seguito e non senza una certa riluttanza.

## MIKROPASCAL PRO PER PIC

Il MikroPascal in sintesi è il linguaggio Pascal (più correttamente dovremmo parlare di ambiente di sviluppo) per PIC Micro. Come MikroC e MikroBasic consente di sviluppare firmware per controllori PIC sia della famiglia mid-range (PIC16F e PIC 12F) che della più evoluta famiglia PIC18. In sintesi possiamo assumere come vera l'equivalenza MikroPascal = PIC Micro + Pascal.
Qualunque algoritmo implementato in MikroPascal deve essere compilato nel linguaggio che il controllore è in grado di comprendere cioè nel linguaggio macchina del controllore (composto unicamente da sequenze di 0 e 1). L'alternativa all'impiego di un linguaggio ad alto livello è l'impiego di un linguaggio a basso livello, in pratica l'assembler, che è caratterizzato da un'interazione più diretta e profonda con l'hardware del controllore e nella quale appare più evidente l'architettura di quest'ultimo; si tratta tuttavia di un linguaggio che per sua natura è di difficile gestione da parte del programmatore. In altre parole, anche nella programmazione dei microcontrollori e dei sistemi embedded vale la considerazione generale secondo cui un linguaggio a basso livello, essendo spostato verso l'hardware, si presta poco a uno sviluppo semplice, lineare e rapido dell'applicazione. Ne è prova il fatto che mentre agli albori della diffusione dei controllori questi erano prevalentemente programmati in assembler, oggi nella quasi totalità dei casi, il programmatore fa uso di linguaggi ad alto livello.
Mentre per poter programmare in linguaggio assembler è necessaria una approfondita conoscenza dell'architettura e dell'hardware del controllore, programmando con un linguaggio ad alto livello è possibile, almeno in parte, ignorare o comunque non conoscere in maniera approfondita tali dettagli. D'altro canto lo sviluppo dei microcontrollori ha visto questi componenti negli anni dotarsi di risorse fisiche sempre più ampie (sempre maggiore quantità di memoria flash e memoria RAM integrata, sempre crescente numero di pin e di periferiche di I/O gestibili). Questa evoluzione ha visto crescere progressivamente anche le dimensioni e la complessità del codice sviluppato per piattaforme a microcontrollore. Proprio per questo, la necessità di disporre e impiegare linguaggi ad alto livello è divenuta una prerogativa praticamente irrinunciabile.
La necessità di passare dall'impiego dell'Assembly a quello di linguaggi ad alto livello ha seguito in qualche modo lo stesso naturale percorso che negli anni Sessanta e Settanta ha visto l'affermazione dei linguaggi di programmazione nel modo dei calcolatori.
Così come quarant'anni fa si è avuto lo sviluppo di linguaggi di programmazione per PC quali Basic, Pascal e C negli ultimi dieci-quindici anni, in particolare, si è avuto un forte sviluppo ed una forte diffusione di analoghi linguaggi nell'ambito dei microcontrollori.
Il MikroPascal è al tempo stesso un IDE, un linguaggio ad alto livello e un compilatore per controllori PIC. Sia Microchip sia software-house terze hanno sviluppato compilatori per PIC. Il MikroPascal PRO, come anche MikroC PRO e MikroBasic PRO (tutti e tre sviluppati e prodotti dall'azienda serba Mikroelektronika) sono un esempio di compilatori per PIC alla pari di altri quali CCS ed Hi-Tech.
A proposito di questi, e in particolare del MikroPascal, è doveroso precisare che non si tratta di prodotti gratuiti e che la casa produttrice, come abbiamo già avuto modo di segnalare, mette a disposizione, gratuitamente, la sola versione demo. Quest'ultima è limitata nelle dimensioni massime del codice compilato ma a parte questa limitazione è perfettamente funzionante e consente la scrittura e la compilazione di codice per PIC delle famiglie PIC12, PIC16 e PIC18.
In sintesi, le principali caratteristiche di MikroPascal:
- si presenta come IDE (*Integrated Development Environment*) integrante, in un unico pacchetto, un editor di testo per la stesura del programma sorgente, un compilatore Pascal per PIC, un debugger e un programmatore per il caricamento del firmware compilato sulla memoria flash del PIC;
- integra una serie di tool e facility tra cui un convertitore in notazione HEX, BIN e DEC, un Code Explorer avanzato, un ottimo assistente di codice, un help on-line molto ben curato;
- mostra il codice Assembler del programma;
- consente una gestione efficace del listato di programma all'interno dell'editor, grazie alla colorazione del codice sorgente, in accordo con il riconoscimento della sintassi del codice.
- consente la compilazione del codice sorgente cioè la generazione del file HEX da caricare sulla memoria flash del PIC;
- integra strumenti statistici sotto forma di report grafici che quantificano il livello di sfruttamento delle risorse fisiche del controllore oggetto dell'applicazione firmware che si sta scrivendo;

# Firmware

**l'unica rivista italiana nativa digitale per i professionisti dell'elettronica e dedicata, principalmente, ai microcontrollori, dispositivi FPGA, componentistica analogica e approfondimenti sulle tecnologie**

- non più solo testo ma anche video!
- possibilità di grande interazione per il lettore
- links a tutte le varie risorse aggiuntive
- possibilità di cercare un testo nella rivista corrente e nell'archivio (!)
- possibilità di stampare tutta la rivista o anche solo alcune parti
- possibilità di leggere la rivista offline scaricandola sul PC
- possibilità di leggere la rivista con gli e-reader (compreso iPhone e iPad)
- moltissime riviste in archivio GRATIS per i nuovi abbonati
- abbonamento a partire da € 19.50
- Possibilità di scaricare la rivista in pdf (solo per gli abbonati)

Richiedi la tua copia omaggio qui:
***http://mailing.fwonline.eu***

CODICE MIP 2827163

• presenta un costo accessibile e un ottimo rapporto qualità/prezzo;
• è disponibile anche in versione demo gratuita con l'unica limitazione di consentire la generazione di compilati non più lunghi di 2K (versione scaricabile dal sito di Mikroelektronika www.mikroe.com) più che sufficienti per chi voglia cimentarsi nello studio dei microcontrollori PIC12, PI16 e PIC18 e nell'apprendimento del linguaggi di programmazione MikroPascal per PIC o sviluppare applicazioni non particolarmente complesse (alcuni microcontrollori PIC di fascia bassa presentano memoria inferiore a 2K e pertanto non potrebbero nemmeno ospitare firmware di dimensioni maggiori).
Queste caratteristiche rendono il pacchetto un prodotto di fascia professionale comunque avvicinabile anche da appassionati e hobbisti. La disponibilità della versione demo è un aspetto particolarmente apprezzato da questi ultimi. Il suo successo è testimoniato dal fatto che la casa produttrice ne ha proposto nel tempo continui aggiornamenti arricchiti via via di nuove librerie di funzioni proponendo parallelamente revisioni sempre nuove di proprie demo-board. Su queste avremo modo di dare alcuni cenni, al momento opportuno, nel prosieguo del corso.
Va detto, per completezza, che Microchip mette a disposizione gratuitamente un proprio ambiente di sviluppo integrato per la programmazione in Assembly (MPLAB) e un simulatore per il debug (MPSIM). All'interno dell'IDE MPLAB è possibile installare come plug-in diversi compilatori.
Diverse software house hanno sviluppato sia compilatori pensati come integrati in propri ambienti di sviluppo sia tali da poter essere integrati all'interno dell'ambiente di sviluppo Mplab di Microchip. I compilatori C per PIC18, PIC24/dsPIC e PIC32 sviluppati da Microchip sono esempi appartenenti alla seconda categoria, MikroBasic, MikroC e MikroPascal sono invece esempi di compilatori appartenenti alla prima.

## IL NOSTRO BANCO DI LAVORO

Quando si scrive un firmware per un microcontrollore è certamente necessario disporre di un ambiente di sviluppo per scrivere e compilare il codice ma è anche necessario disporre di qualcosa che consenta di testare il corretto funzionamento del codice stesso, ovvero la rispondenza di questo all'algoritmo che si intendeva implementare. A tale scopo è possibile agire in tre diversi modi:
1) disporre di una breadboard su cui realizzare fisicamente il circuito a microcontrollore e un programmatore opportunamente interfacciato alla stessa per poter riversare il codice compilato sulla memoria flash del controllore;
2) disporre di una demo-board o di una scheda di sviluppo appositamente pensata allo scopo. Le schede di sviluppo *EasyPIC* costituiscono un esempio di demo-board. Si tratta indubbiamente di un'opzione comoda e molto flessibile dal momento che tali demo-board integrano molto dell'hardware con il quale il nostro controllore PIC può trovarsi a interfacciarsi. Dispongono di predisposizioni hardware per il collegamento di altri dispositivi esterni e integrano, quasi sempre, un programmatore USB on-board;
3) disporre di un simulatore software di PIC che consenta di simulare, in maniera visuale, componenti quali diodi, display, potenziometri, pulsanti e, più in generale, porte di comunicazione.
La soluzione 1) è più adatta a coloro che conoscono già il mondo dei microcontrollori e che hanno un minimo di esperienza pratica nella realizzazione di prototipi di circuiti elettronici.
La 2) è invece una soluzione ottimale un po' per tutti: evita a coloro che sono alle prime armi di dover affrontare lo scoglio della realizzazione fisica della scheda e facilita il lavoro di quanti, pure essendo esperti in materia, necessitano di uno sviluppo rapido delle applicazioni che privilegi l'aspetto firmware e svincoli il più possibile, almeno inizialmente, dall'onere di realizzare una board. In ambito professionale è ormai l'approccio maggiormente utilizzato soprattutto in fase di prototipazione dell'applicazione.
La terza possibilità è l'unica che consente di apprendere e studiare l'argomento, sperimentando pur non disponendo e non realizzando alcunché di fisico. È sufficiente un PC, il compilatore (MikroPascal in questo caso) e il software di simulazione. Inutile sottolineare che quest'ultimo approccio consente di studiare e sperimentare ma ovviamente non consente di utilizzare fisicamente l'applicazione.
Al fine di rendere più semplice e flessibile l'apprendimento, nel corso delle puntate, ci orienteremo verso gli approcci descritti ai punti 2) e 3), convinti che sperimentare sia comunque il miglior modo per imparare. Questo non impedisce a coloro che hanno sufficiente dimestichezza pratica con la realizzazione dei PCB e schede elettroniche di realizzare i circuiti a microcontrollore che proporremo e di utilizzare un programmatore per la programmazione della memoria flash del micro.

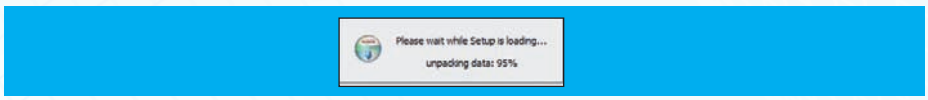



*Figura 1: avvio dell'installazione.*

*Figura 2: messaggio di benvenuto e invito a chiudere le applicazioni aperte.*

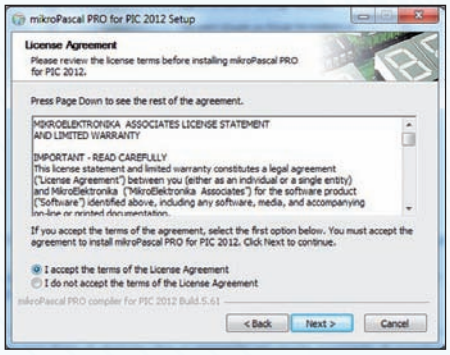



*Figura 3: accettazione dei termini di licenza.*

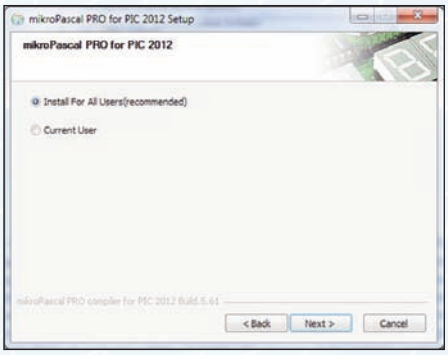



*Figura 4: scelta dell'installazione per l'utente corrente (current user) o per tutti gli utenti (Install for All Users).*

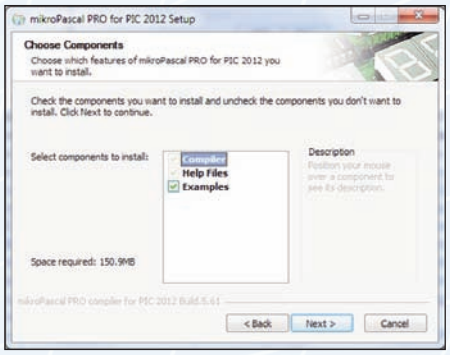



*Figura 5: i file del compilatore e quelli della guida in linea vengono installati di default mentre è possibile scegliere se installare o meno i file di esempio (l'installazione è consigliata).*

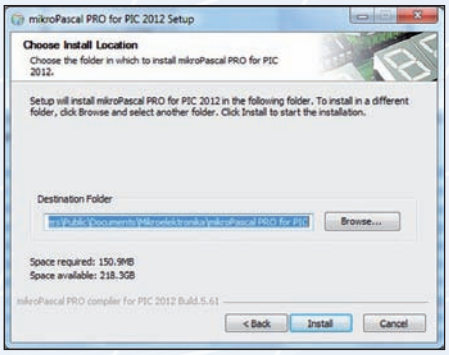



*Figura 6: impostazione del percorso di installazione.*

## L'IDE MIKROPASCAL

Fatte le doverose premesse ed inquadrato il corso che ci accingiamo a sviluppare, entriamo nel merito dell'ambiente di sviluppo MikroPascal. Per meglio comprendere quanto esporremo, si consiglia al lettore di eseguire il download gratuito della versione demo dell'ambiente di sviluppo MikroPascal direttamente dal sito del produttore (Mikroelektronika, sito web: www.mikroe.com) e di installarlo sul proprio PC.
Il pacchetto non risulta particolarmente pesante e come detto precedentemente presenta come unica limitazione quella di consentire la generazione di codice compilato di dimensione non superiore a 2k. Al momento di scrivere questo articolo l'ultima versione di MikroPascal rilasciata sul proprio sito da Mikroelektronika e la nuova versione 5.61 cui faremo riferimento da questo momento in avanti.

## INSTALLAZIONE DI MIKROPASCAL

Il file eseguibile (.exe) di installazione (mikroPascal_pro_PIC_2012_Build.5.61) è contenuto in un file .zip (mikropascal_pro_pic_2011_v561). Avviata l'installazione **(figura 1)** con l'esecuzione di detto file, un form di dialogo fornisce un messaggio di benvenuto e consiglia di chiudere le applicazioni aperte prima di continuare **(figura 2)**.
La pressione del pulsante *Next* consente di proseguire nell'operazione di installazione. Il form che compare in successione chiede l'accettazione dei termini della licenza d'uso **(figura 3)**.
Letti e accettati i termini della licenza (*I accept the terms of…)* è possibile proseguire nell'installazione attraverso il pulsante *Next*. E necessario, a questo punto, scegliere se installare il pacchetto per l'utente corrente o per qualunque utente. Lasciando spuntata la seconda opzione (scelta consigliata) si prosegue oltre con *Next* ***(figura 4)***.
Il nuovo form di dialogo **(figura 5)** che si presenta all'utente invita a scegliere i componenti del pacchetto che si intende installare. Sicuramente è necessario installare il compilatore e i file dell'help che appaiono già spuntati e che non possono essere deselezionati, mentre è possibile scegliere se installare o meno i file di esempio. In ogni caso è opportuno installare anche questi ultimi poiché risulteranno di estrema utilità durante la stesura delle applicazioni.
Attraverso il pulsante Next si accede alla finestra di dialogo all'interno della quale specificare la cartella in cui installare il pacchetto (*destination folder*, **figura 6**). Generalmente non vi sono motivi per installare il pacchetto in un percorso differente da quello di default proposto dall'installer.
La pressione del pulsante *Install* avvia l'installazione, che consiste essenzialmente in un processo di copia automatica di tutti i file e degli eseguibili componenti il pacchetto all'interno del percorso precedentemente specificato.
La progress bar mostrata a video consente di seguire l'avanzamento dell'installazione. Attraverso il pulsante Show details è possibile visualizzare il dettaglio di tutti i file che vengono progressivamente estratti e installati nel percorso di installazione **(figura 7)**.
Al termine del processo compare un form di dialogo che avvisa che l'installazione è andata a buon fine e che invita a terminare la stessa attraverso il pulsante Finish. L'installer chiede se si desidera installare l'applicativo integrato per la programmazione della flash del controllore (mikroProg Suite v. 2.20). Scegliendo l'opzione "SI" **(figura 8)** , si avvia una procedura per l'installazione di mikroProg Suite, operazione che evolve in maniera del tutto simile a quella appena conclusa per MikroPascal.
A questo punto il pacchetto di installazione è installato sia per quanto riguarda l'IDE MikroPascal sia per quanto riguarda l'applicativo di programmazione della flash **(figura 9)**; è possibile quindi eseguire il primo avvio di MikroPascal (**figure 10** e **11**). Al momento di scrivere questo articolo si è installato il pacchetto sia su un sistema Windows XP sia su sistema Windows 7. In entrambi i casi, la procedura descritta ha portato alla corretta installazione del pacchetto. Le due icone per l'avvio di *MicroPascal Pro for PIC* e *microProg Suite for PIC* appaiono disponibili sul desktop **(figura 10)**.

## STRUTTURA E ORGANIZZAZIONE DELL'IDE

La comparsa dell'IDE, una volta avviato il pacchetto, secondo la procedura descritta al termine del precedente paragrafo, è preceduta da uno splash screen che riporta alcune informazioni su MikroPascal, tra cui la versione. In **figura 12** è mostrata la schermata dell'IDE così co-

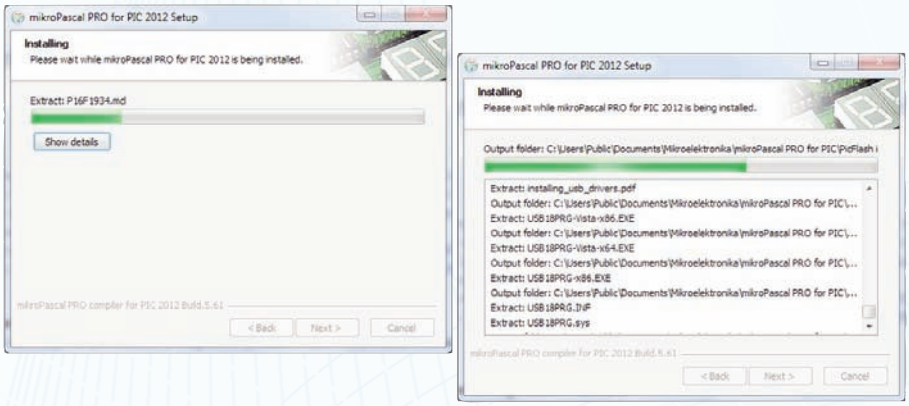



*Figura 7: installazione in corso. Attraverso il pulsante "Show details" è possibile visualizzarne i dettagli.*

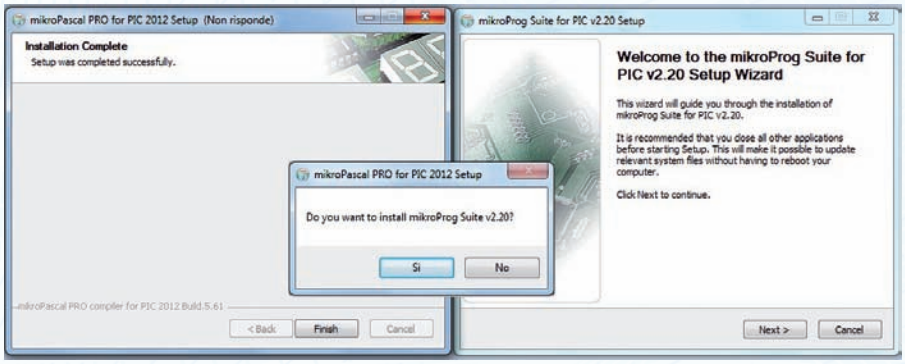



*Figura 8: fine dell'Installazione di MikroPascal e avvio dell'installazione di mikroProg Suite v. 2.20.*

me appare subito dopo il primo avvio. Rispetto a versioni precedenti del software, appare evidente la zona centrale della schermata che riporta un serie di avvi rapidi di attività pensate per coloro che non conoscono affatto l'ambiente di sviluppo MikroPascal (*Welcome to MikroPascal PRO for PIC, Get Started with MikroPascal for PIC, Register your MikroPascal for PIC*). Si tratta della scheda *Get Started*. In alto a sinistra appare invece la scheda *Start Page* che consente di creare un nuovo progetto (*New Project*), aprire un progetto preesistente (*Open Project*) oppure aprire progetti di esempio (*Open Exemples Folder…*). Nelle prossime puntate vedremo più compiutamente come creare un nuovo progetto e come utilizzare i progetti di esempio. Al momento ci limitiamo ad analizzare brevemente le diverse aree dell'IDE. Al fine di rendere più efficace la nostra panoramica apriamo un progetto tra quelli di esempio installati precedentemente.

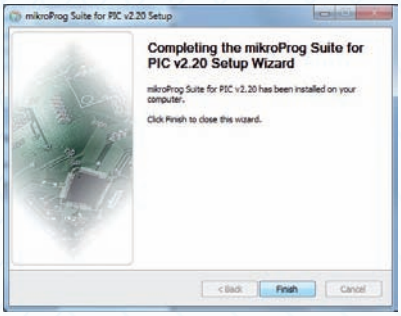



*Figura 9: fine dell'installazione di mikrProg Suite 2.20.*

### Code Explorer

Compare a sinistra, è accessibile attraverso l'omonimo tab laterale **(figura 13)** e, come il suo nome indica esplicitamente, consente di eseguire una vera e propria esplorazione del codice. Esso mostra, infatti, in una struttura ad albero, i nomi delle procedure e delle funzioni componenti il codice oltre a quelle delle variabili globali e locali. L'analisi di questo albero consente di avere una visione immediata della struttura del codice e degli elementi in esso inseriti quali funzioni, variabili, costanti oltre che il loro valore e il loro àmbito di visibilità **(figura 14)**. Se si esegue un clic su una delle voci del Code Explorer, automaticamente quell'elemento viene evidenziato all'interno del codice nella posizione in cui è inserito. Questo consente di trovare rapidamente gli elementi nel codice e aiuta a evitare errori nell'assegnazione dei nomi alle variabili e alle funzioni.

*Figura 10: a installazione completata, sul desktop compaiono le due icone di avvio di MicroPascal e microProg Suite.*

### Project Settings

Attraverso questa scheda, accessibile attraverso l'omonimo tab laterale **(figura 13)**, è possibile scegliere i microcontrollore per il quale scrivere il codice, la frequenza di clock del microcontrollore, che corrisponde in hardware con quella impostata attraverso un circuito con oscillatore al quarzo connesso al controllore sulla porta di oscillazione, e la tipologia di debugger **(figura 15)**. Il modello di microcontrollore e la frequenza di clock in altri compilatori vengono indicati attraverso l'inserimento di opportune direttive nel testo del codice sorgente; la disponibilità del *Project Setting* invece rende più agevole questa operazione all'utente di MikroPascal.

### Code Editor

È la parte più ampia dell'IDE MikroPascal ed è l'area all'interno della quale il programmatore scrive il codice **(figura 13)**. Altro non è che un editor di testo che integra alcune semplici ma utilissime funzionalità di correzione e controllo del testo. In particolare il Code Editor integra funzioni di *Code Assistant* **(figura 16)**. Si tratta di funzionalità che facilitano e velocizzano la scrittura del codice riducendo al tempo stesso la probabilità di commettere errori di digitazione. Se si deve digitare un comando (sarebbe più giusto dire "una parola chiave"), per esempio, è sufficiente scriverne i primi caratteri e premere contemporaneamente i tasti <CTRL> e <SPAZIO>; compare una finestra con la lista di tutti i possibili comandi inizianti con i caratteri digitati. A questo punto è sufficiente scorrere la lista

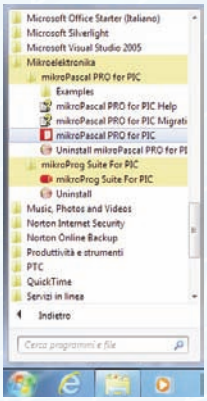



*Figura 11: MikroPascal e mikroProg possono essere avviati anche da "Tutti i programmi\Mikroelektronika\...".*

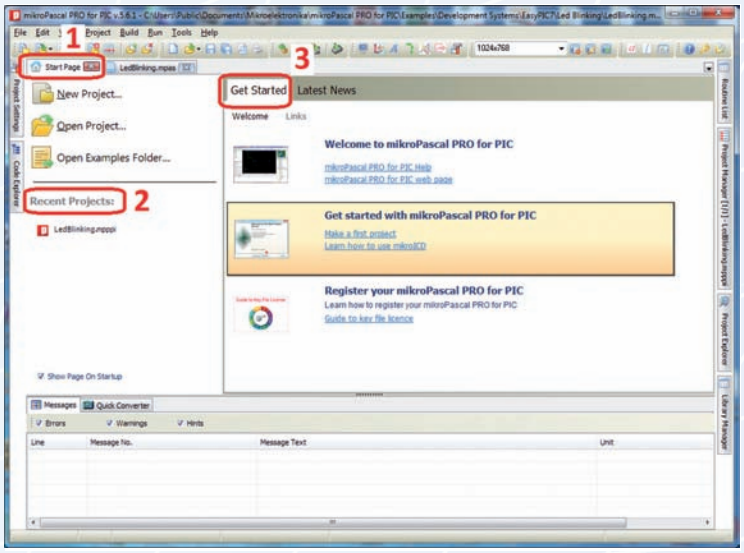



*Figura 12: l'IDE MikroPascal così come si presenta all'utente al primo avvio.*

fino a posizionare il puntatore del mouse sul comando desiderato e premere INVIO per il suo inserimento. Una funzionalità simile, che aiuta il programmatore a scrivere i nomi delle funzioni e i parametri da passare alle stesse funzioni, è quella comunemente detta *Parameter Assistant* (cui si accede dall'interno del Text Editor attraverso Ctrl+Shift+Spazio). Il codice scritto nel Code Editor viene anche automaticamente "colorato" in maniera tale che sia possibile distinguerne facilmente le diverse entità che compongono il listato di programma cioè istruzioni, commenti, costanti, tipi, ecc.). Riuscire a distinguere facilmente e a colpo d'occhio queste differenti entità aumenta infatti la leggibilità e la comprensione del codice, ne facilita la modifica in corso d'opera e consente di rendere più evidenti eventuali errori di scrittura.

### Library Manager

È una scheda dell'IDE **(figure 13 e 17)** che consente di selezionare le librerie da inserire nel progetto. È un'indicazione essenziale per il compilatore che ha bisogno di conoscere quali siano le librerie da compilare insieme con il codice Pascal che il programmatore ha scritto nel Code Editor, per generare correttamente il codice eseguibile.

### Rountine List

È una scheda accessibile attraverso il tab omonimo **(figure 13 e 18)** che riporta l'elenco delle funzioni disponibili e consente di accedere alla loro collocazione nel codice. È sufficiente cliccare sul nome della funzione nell'elenco Runtine List per accedere direttamente alla parte di codice relativa.

### Messages/QConverter

È l'area localizzata nella parte bassa dell'IDE e integra due distinte schede ciascuna delle quali attivabile mediante il relativo e omonimo tab **(figure 13 e 19)**. La scheda *Messages*, accessibile attraverso l'omonimo tab, è una finestra riservata alla notifica di messaggi generali, di errore o di warning generati dal compilatore MikroPascal durante le operazioni di compilazione e/o di linking del codice sorgente. Il processo di compilazione e quello di linking possono infatti anche non andare a buon fine, nel qual caso il compilatore notifica uno o più messaggi di errore che possono rivelarsi utili nella correzione del codice. In questo caso il file .HEX non viene generato. Nel caso in cui il processo giunge a termine invece potrebbero essere segnalati dei warning che non necessariamente costituiscono evidenza di errori.
Sta pertanto al programmatore valutare adeguatamente la segnalazione ricevuta. In ogni caso, anche quando il processo di compilazione giunge correttamente a termine e il file .HEX viene correttamente generato, nulla può assicurare che il codice compilato esprima effettivamente e senza errori l'algoritmo che il programmatore intendeva implementare.
La scheda *Quick converter* invece **(figura 19)** altro non è che un convertitore

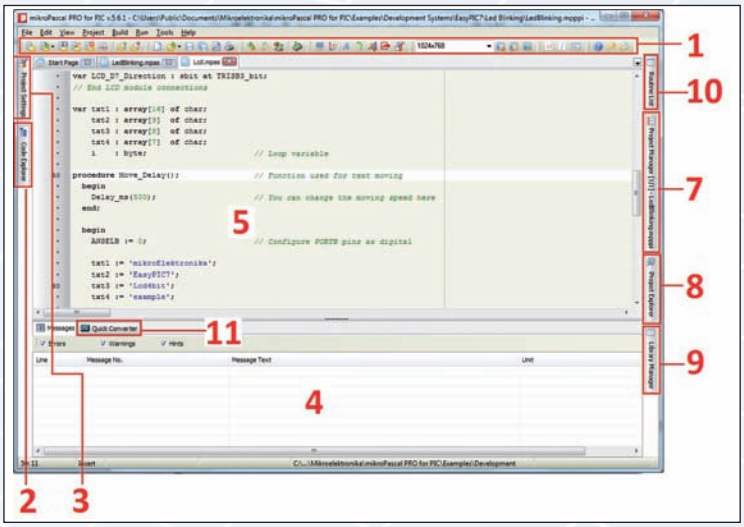



*Figura 13: parti componenti l'IDE integrato di MikroPascal (1-Menu Toolbar, 2-Code Explorer, 3-Project Settings, 4-Messages, 5-Code Editor, 7-Project Manager, 8-Project Explorer, 9-Library Manager, 10-Runtine List, 11-Quick Converter).*

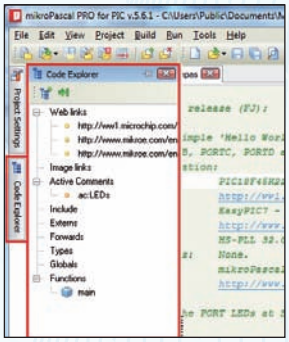


*Figura 14. Code Explorer (struttura ad albero del programma).*
*Figura 15: scheda Project Settings.*

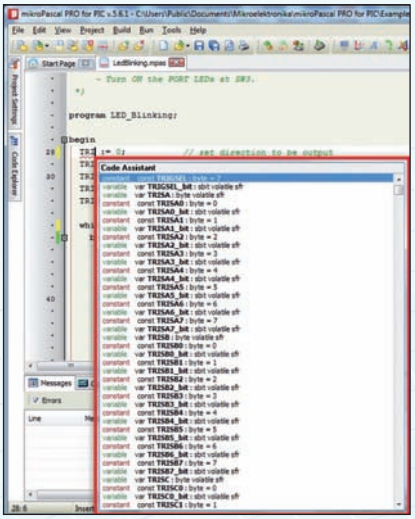
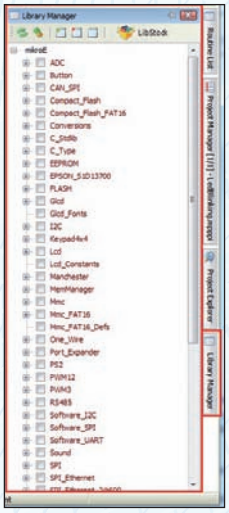
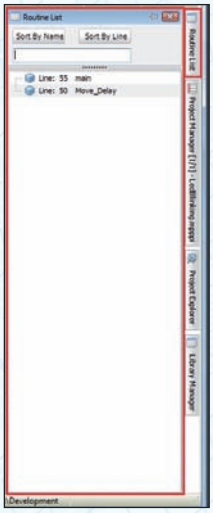


*Figura16: Code Assistant.*

*Figura 17: Library Manager.*
*Figura 18: Runtine List.*





integrato che consente di eseguire rapidamente conversioni tra valori decimali, esadecimali e binari. Si tratta di operazioni abbastanza ricorrenti nel corso della stesura di un codice di programma.

### Project Explorer

Consente di esplorare tutti i progetti tra cui quelli di esempio presenti sul PC **(figura 20)**. Questo facilita l'accesso ai progetti di esempio e pertanto il loro utilizzo durante lo sviluppo delle applicazioni.

### Project Manager

Accessibile attraverso l'omonimo tab, consente la gestione dei progetti multipli **(figura 21)**.

## CONCLUSIONI

In questa prima puntata del corso MikroPascal per PIC abbiamo presentato l'ambiente di sviluppo MikroPascal inquadrandolo all'interno del più ampio panorama dei controllori PIC di Microchip. Attraverso il download della versione demo del pacchetto MikroPascal (ver. 5.61) e la sua installazione ci siamo dotati dello strumento base per affrontare il nostro viaggio nella programmazione Pascal dei controllori PIC.

## NELLA PROSSIMA PUNTATA

Nella prossima puntata approfondiremo la conoscenza dell'IDE MikroPascal analizzando una serie di utility e di strumenti integrati nell'ambiente di sviluppo che aiutano lo sviluppatore a scrivere il codice di programma e a tenerlo costantemente sotto controllo. Impareremo quindi a utilizzare l'ambiente di sviluppo MikroPascal per la creazione del nostro primo progetto. In tale occasione, dovendo imbatterci in concetti quali memoria flash, memoria RAM, e I/O (Input/Output), forniremo alcuni cenni all'architettura dei controllori PIC per coloro che sono alle prime armi con il mondo dei microcontrollori e dei PIC in particolare. ❑

**CODICE MIP 2827071**

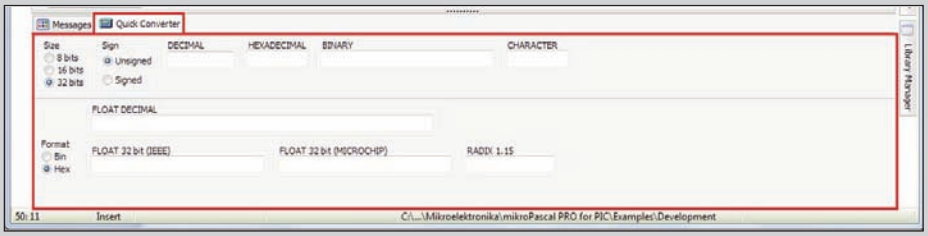


*Figura 19: scheda Quick Converter (conversioni di codici, decimali, esadecimali e binari).*

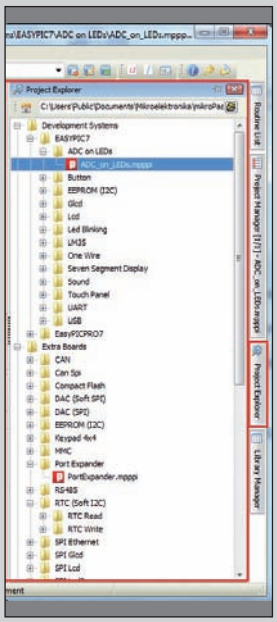
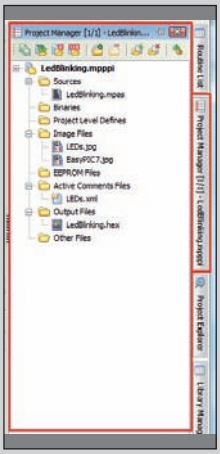


*Figura 20: scheda Project Explorer.*

*Figura 21: Project Manager.*



## PER *approfondire...*

- Linguaggi di programmazione: principi e paradigmi, 2a ed, Milano, McGraw-Hill, 2011-Maurizio Gabbrielli e Simone Martini;
- Informazioni varie sui compilatori di Mikroelektronika (dal sito www.mikroe.com);
- MikroPascal PRO for PIC v.5.61 demo e documentazione relativa (download dal sto www.mikroe.com);
- Help in linea MikroPascal Pro for PIC v. 5.61.

## Una CPU più veloce e con più memoria?
## E' arrivato il nuovo CUWIN5000!

# CUWIN

CODICE MIP 2827207

## Se hai bisogno di funzionalità sofisticate prova il CUWIN, un Touch controller rivoluzionario.

Programmabile in C++ o mediante un qualsiasi linguaggio .Net (C#, VB.Net, etc) per la creazione di applicazioni sofisticate che comprendono acquisizione dati, controllo di macchine, monitoraggio di sensori e addirittura reportistica web, solo per citarne alcune.

Più economico di un PC, e con maggiori prestazioni di un controller HMI Touch, prova il CUWIN, un touch controller rivoluzionario già utilizzato in migliaia di applicaizoni in giro per il mondo.

▶ CUWIN5200 ▶ CUWIN5500 ▶ CUWIN6300

- Samsung 2450 (533MHz) ARM Processor
- 128MB SDRAM / 128MB NAND Flash
- Windows® Embedded CE 6.0
- .Net Compact Framework 3.5
- Software Development: C++ or any .Net Language (C#, VB.Net, etc...)
- Ethernet 10/100Base-TX
- RS-232 x 3, RS-485 x 1
- USB 2.0 x 1, USB 1.1 x 1
- Audio (Line Out) / RTC
- Keyboard and Mouse (optional)
- CE, FCC, and KCC certified

elettroshop.com
brilliant electronics since 1998

via Giotto 7
20032 Cormano (MI)
tel. 02 66504794
info@elettroshop.com

www.elettroshop.com

Interfacciamento dei processori *(parte decima)*

# EFFETTO BLANKING *sui display a due digit*

**I processori, sia come parte integrante dei microcontrollori sia come oggetti programmabili indipendenti, elaborano informazioni esclusivamente binarie: prendiamo in considerazione l'opportunità di visualizzarle, nella nativa forma esadecimale, direttamente su display a due o più digit.**

Nelle puntate precedenti abbiamo fornito le migliori idee per la gestione di uno o due digit direttamente collegati sulla porta d'uscita di un qualunque dispositivo programmabile (microcontrollore o personal computer) oppure attraverso uno strato di interfaccia costituito da una o più memorie esterne a 8 bit di tipo 74LS374 (8 flip-flop D-Type in parallelo) o da componenti appositamente progettati (i decoder da binario a "sette segmenti") per fornire il miglior servizio ai visualizzatori.
In particolare abbiamo apprezzato la possibilità di servire due digit alla volta con un solo byte, alimentando con i suoi due nibble le rispettive quaterne di linee d'ingresso dei decoder chiamati a servirli, i TTL 74LS47 e 74LS48 e il CMOS 4511, sottolineando per altro che essi sono in grado di interpretare solo le prime 10 delle 16 combinazioni possibili di ciascuna quaterna di bit, cioè sono adatti ad decodificare solo il Codice BCD, come noto costituito da 10 elementi, associati ai numeri da 0 a 9 del sistema di numerazione decimale.
Sfortunatamente in commercio esiste un solo componente in grado di decifrare tutte le 16 combinazioni, cioè in grado di interpretare un Codice Binario Puro a 4 bit e di mostrare tutti i 16 simboli del sistema di numerazione esadecimale; si tratta del TTL 9368, da tempo quasi introvabile nei negozi di elettronica ma comunque ancora reperibile su Internet.
Nonostante questa importante differenza funzionale, si tratta di un componente pin-out compatibile (escludendo tre piedini di controllo) e molto simile al CMOS 4511, e (sebbene strutturalmente molto meno sofisticato) anche al TTL 74LS48: tutti e tre hanno 7 linee d'uscita attive alte, adatte a pilotare i segmenti di un digit a catodo comune.
La **figura 1a** mostra il suo pinout e ne evidenzia la struttura logica; appare subito evidente che si tratta di un componente speciale, in grado di dare molto di più di quanto ci si aspetti, cioè di svolgere tre funzioni in una:
a) consente di congelare in una memoria a 4 bit le informazioni fornite sulle linee d'ingresso;
b) garantisce ovviamente il suo intrinseco servizio di Decoder, ora estesa a tutte e 16 le possibili combinazioni;
c) è dotato di driver su ciascuna delle 7 linee d'uscita, al fine di fornire le migliori condizioni di pilotaggio ai LED a esse collegati.
Prima di raggiungere il Decoder ciascuna delle 4 linee d'ingresso attraversa un flip-flop di tipo D-Latch; l'utilizzo di questo tipo di memoria non è molto diffuso nel progetto digitale e questo è uno dei casi più peculiari; i suoi 4 elementi sono attivati sul livello della linea di sincronismo (LE, Latch Enable, pin3), che li controlla contemporaneamente: quando LE è a livello 0, fissato al negativo dell'alimentazione, i 4 bit del Codice BCD applicato in ingresso passano inalterati a valle della memoria, che risulta essere "trasparente" nei loro confronti, così come fosse una porta aperta (l'uscita insegue l'ingresso); nell'istante in cui LE passa da un livello basso a uno alto (cioè sul suo fronte di salita) le uscite D-Latch "scattano" sul codice binario presente su di esse in quel momento, tenendolo bloccato (memorizzandolo) per tutto il tempo nel quale la li-

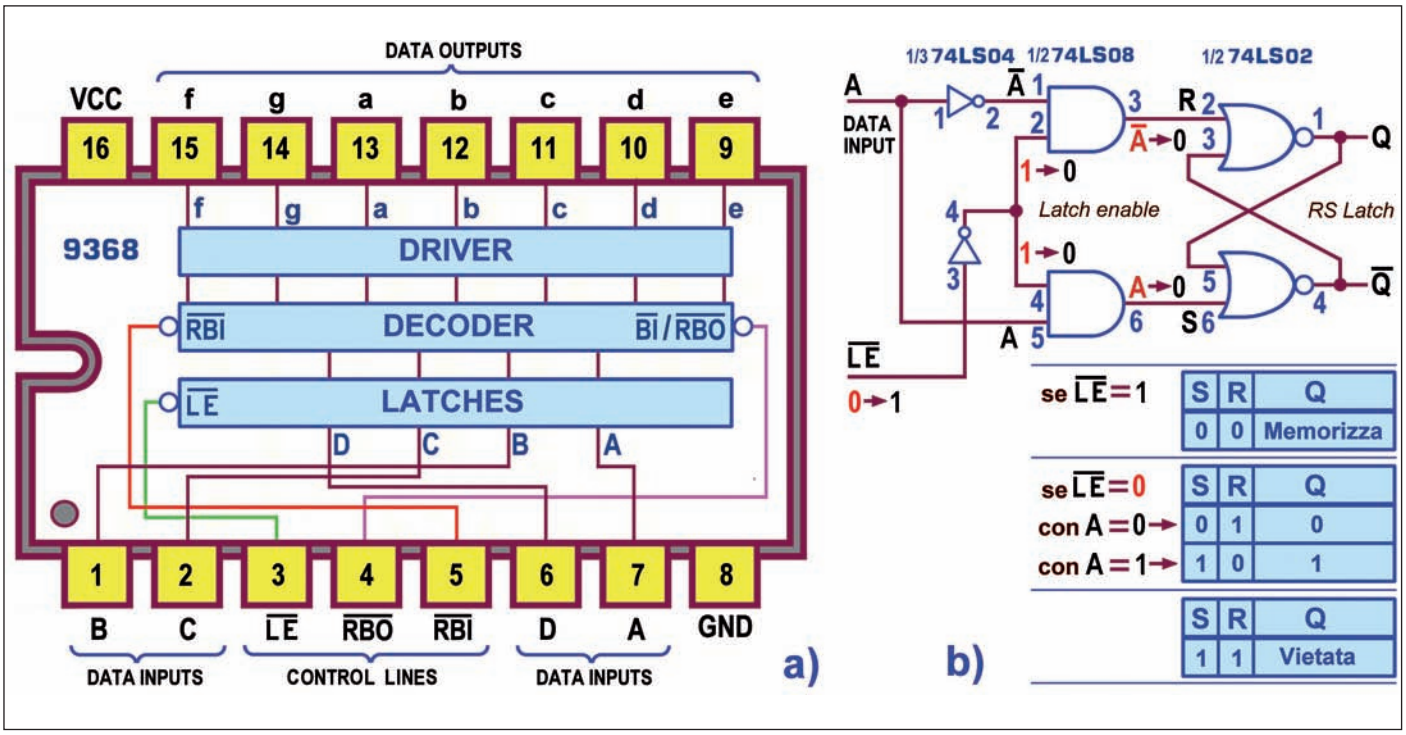



*Figura 1: binary to 7-segment Latch/Decoder/Driver 9368: a) pinout; b) funzionamento logico.*

nea LE è tenuta a livello 1, fino a quando verrà riportata a 0.
La notazione logica mette in evidenza queste specifiche (graficamente, con la presenza del pallino sulla linea d'ingresso e formalmente soprasegnando il segnale LE): in condizioni di funzionamento normale (oppure se l'azione della memoria non è desiderata) questo pin3 va posto a massa! Si tratta di uno dei più frequenti errori dei tecnici inesperti che, dimenticandolo scollegato, si disperano perché il digit controllato dal Decoder rimane "bloccato" anche quando è attivo un eventuale conteggio in ingresso.
Va detto che la notazione attiva bassa per LE può generare qualche insicurezza se la si intende come comando per "eseguire la memorizzazione" degli ingressi: di fatto essa avviene quando LE è a 1 così da poter supporre che l'atto si compia per un'azione attiva alta! In realtà il termine Latch Enable va inteso come comando per "autorizzare il passaggio del dato sul latch", lasciando intendere che quest'ultimo sia abilitato a compiere la sua funzione (quella di memorizzare il dato stesso) solo se il segnale LE tiene aperte le porte d'accesso al latch, cosa possibile appunto se esso è forzato a 0, per la presenza di un inverter interno (come si vede dallo schema di **figura 1b**) valido per ciascuno dei 4 flip flop D-Latch utilizzati dal componente.
Come anticipato, unico tra tutti i Decoder conosciuti, questo componente interpreta tutte le possibili 16 combinazioni del Codice binario a 4 bit proposto in ingresso, assicurando per ciascuna di esse il corrispondente simbolo del sistema di numerazione esadecimale; l'aspetto del digit collegato al decoder è dunque quello mostrato in **figura 2**.
La novità consiste nella visualizzazione dei simboli alfabetici da A a F (proposte con lettere minuscole o maiuscole) in aggiunta ai simboli numerici da 0 a 9 (già disponibili con gli altri Decoder); tra l'altro nella visualizzazione del 6=$(0110)_2$ e del 9=$(1001)_2$ è previsto il trattino, rispettivamente in alto o in basso, non attivato negli altri.

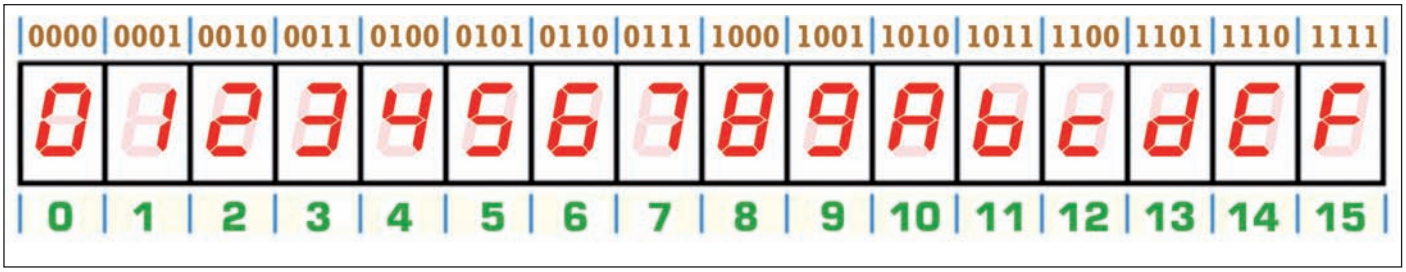



*Figura 2: binary to 7-segment Latch/Decoder/Driver 9368: simboli visualizzati con FND500.*

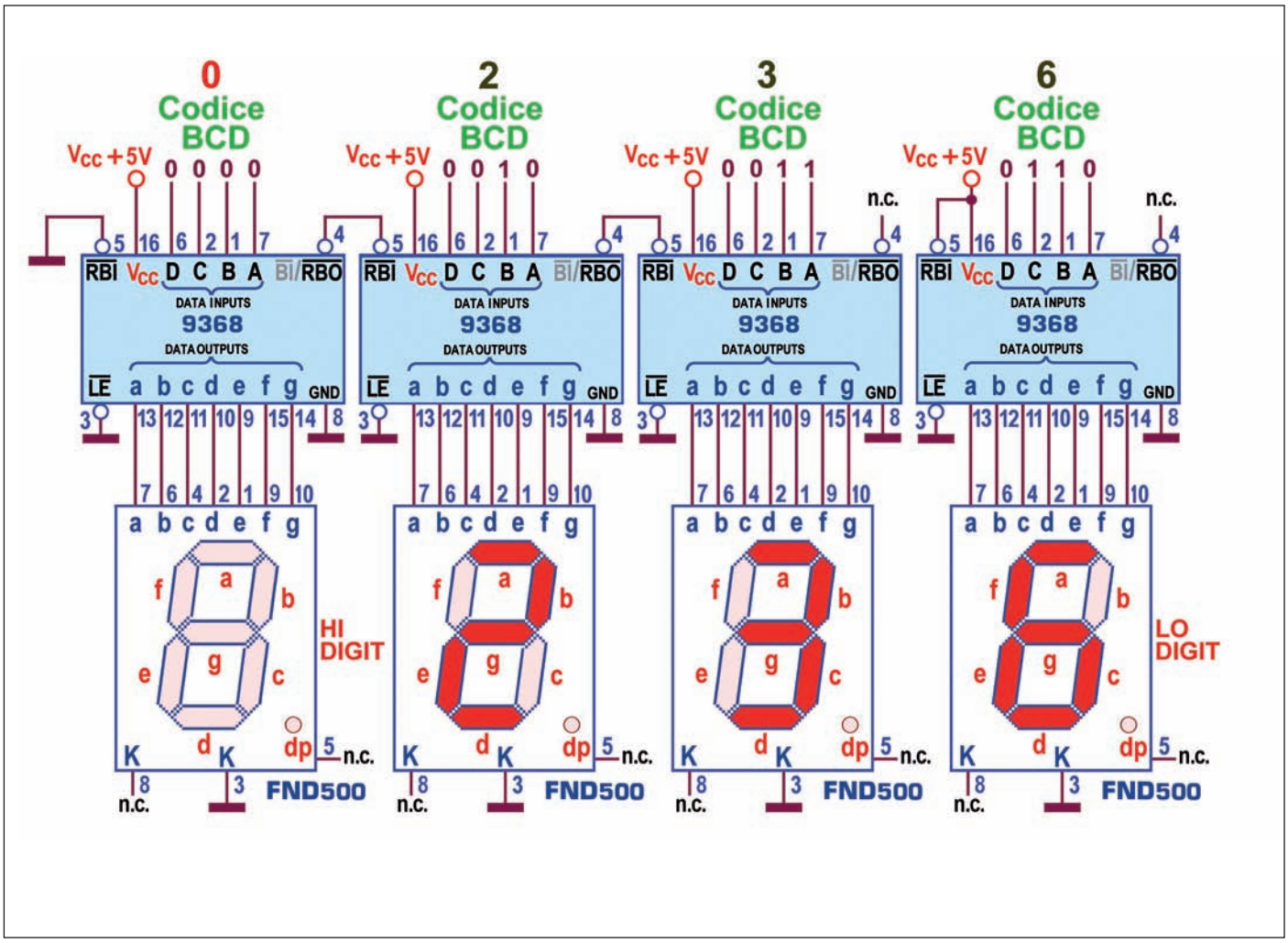



*Figura 3: visualizzatore a 4 digit FND500 con 9368: effetto Ripple Blanking.*

Su questa parte circuitale agisce l'influenza dei segnali BI/RBO (pin4) e RBI (pin5), entrambi attivi bassi, gli stessi presenti nel Decoder TTL 74LS48, usati:

a) BI, Blanking Input: per spegnere incondizionatamente il digit collegato al Decoder, con qualunque codice presente sugli ingressi DCBA;

b) RBO, Ripple Blanking Output e RBI, Ripple Blanking Input: per gestire più componenti 9368 in cascata, con la tecnica di Ripple Blanking (garantita anche dal 74LS48 ma non dal 4511), atta a evitare di accendere tutti gli zero non significativi nell'ambito di un visualizzatore decimale con numerose cifre, sia a sinistra sia a destra (se il numero mostrato è frazionario).

In sintesi il meccanismo di Ripple Blanking funziona così:

a) l'ingresso RBI consente (se attivato, cioè forzato a 0) di spegnere (Blanking) il digit collegato al Decoder solo se contemporaneamente il codice presente sui suoi ingressi DCBA è $(0000)_2$; in queste condizioni forza a 0 anche l'uscita RBO; se il codice presente sugli ingressi DCBA è diverso da $(0000)_2$ il digit mostra regolarmente i simboli corrispondenti e l'uscita RBO rimane disattiva (cioè a livello logico 1);

b) la condizione per poter visualizzare comunque l'eventuale cifra zero è di lasciare l'ingresso RBI scollegato (o meglio collegato al +5 V), cioè disattivo (forzato a livello logico 1);

c) l'uscita RBO serve per reiterare questo servizio anche sul decoder di un eventuale digit meno significativo, posto a destra del primo: collegato all'ingresso RBI del Decoder successivo produrrà lo stesso effetto, tenendo spento il digit a esso collegato solo se sui suoi ingressi DCBA è presente il codice $(0000)_2$;

d) collegando in cascata le uscite RBO di ciascun decoder (a cominciare da quello associato al digit più significativo) con l'ingresso RBI del Decoder successivo (subito a destra) è dunque possibile evitare di accendere tutti gli antiestetici e inutili zero a sinistra di un numero: anche nella pratica comune sembra logico scrivere 236 piuttosto che 0236.

Va peraltro notato che se il 9368 è utilizzato nel pieno della sua specificità (cioè quella di mostrare numeri esadecimali) questo servizio è comunque illogico, sia per il fatto che anche gli eventuali zero a sinistra, nel numero visualizzato, mantengono la loro necessità d'esistere (ba-

# Il primo oscilloscopio mixed-signal low-cost!

Oscilloscopio, Analizzatore di spettro

Analizzatore Logico, Generatore Logico

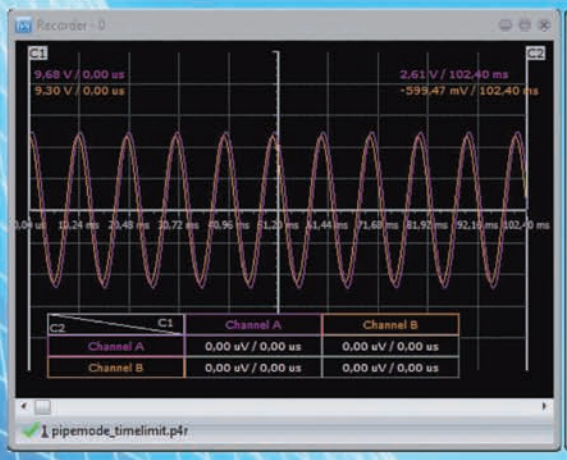
2+5 Canali analogici (12bit, 1MS/s)

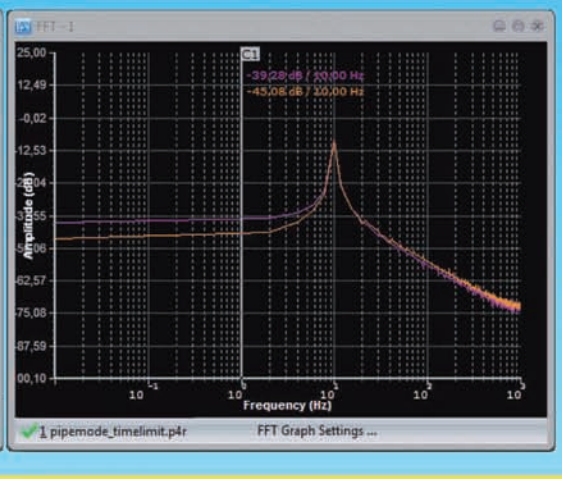
Generatore di onde quadre e PWM

Registratore Digitale

16 Canali Digitali

Aggiornamenti software gratuiti




CODICE MIP 2827211

### Oscilloscopio ed analizzatore di spettro

**Numero canali:** 2+5
**Frequenza di campionamento:** fino a 1MHz
**Memoria:**
- Buffer di lettura: 1126 campioni/canale (1 canale), 563 campioni/canale (2 canali).
- Pipe di lettura: 64K campioni/canale (1 o 2 canali).

**Massima tensione di ingresso:** -20 ÷ +20 V
**Risoluzione ADC:** 12bits
**Triggering:**
- Assoluto (per fronti di salita/discesa)
- Differenziale (per differenza tra campioni consecutivi)
- Esterno (per fronti di salita/discesa di segnali TTL)

**Funzionalità disponibili:** Hamming, Hanning, Blackman, Blackman-Harris.

### Analizzatore logico

**Numero canali:** 16 (8 se utilizzato il generatore logico)
**Frequenza di campionamento:** fino a 10MHz
**Memoria:**
- Buffer in lettura (Fs=4-8 MHz) 128 bit/canale.
- Buffer in lettura (Fs=2-2.66 MHz) 1160 bit/canale.
- Buffer in lettura (Fs<=1 MHz) 1544 bit/canale
- Buffer in lettura (in mod. concatenamento) 1 Mbit/canale.
- Pipe di lettura (Fs < 500KHz) 4K a 256 Mbit/canale.

**Massima tensione di ingresso:** 0 ÷ +5 V
**Triggering:** per fronti del segnale, maschere, impulsi persi, clock esterno.
**Clock:** interno/esterno

### Registratore

**Frequenza di campionamento:** fino a 1MHz
**Capacità massima di registrazione:** 24 ore (Fs < 100 Hz)
**Tensione d'ingresso:** -20 ÷ +20 V (hardware 2 sub-band)
**Risoluzione ADC:** 12bits

### Generatore logico

**Numero canali:** 8
**Frequenza di campionamento:** fino a 1MHz
**Memoria:** 4000 bit/canale
**Tensione di uscita:** "0" - 0 V, "1" - 3.3 V
**Massima corrente in ingress/uscita:** 10 mA

## Novità! Il più economico isolatore USB!

PoScope mega1

*Ordinali subito su www.ieshop.it/poscope*

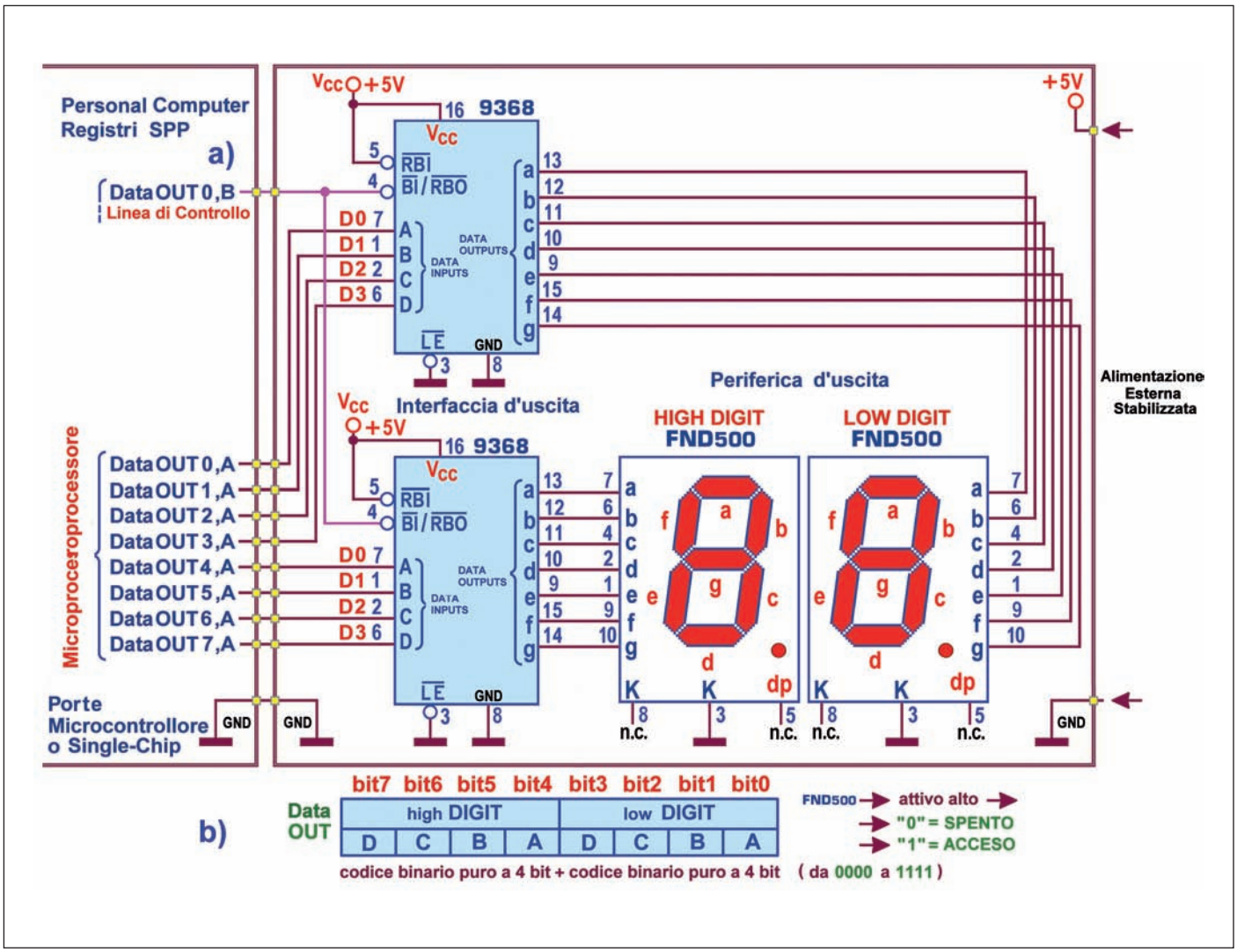



*Figura 4: interfaccia d'uscita con due 9368 (Binary to 7-segment Decoder) per display con 2 FND500.*

sti pensare a un indirizzo o un dato a 16 bit, come 0AF5H o come 001EH) sia per l'improbabilità di dover mostrare un numero esadecimale frazionario, rendendo così inutile anche il potenziale servizio per gli zero a destra.

Ma è pur vero che questo dispositivo si comporta benissimo anche nell'àmbito di un visualizzatore decimale, dando in questo caso per scontato che l'informazione fornita in ingresso ai Decoder sia di tipo BCD, anche se essi sarebbero in grado di processare tutte le 16 possibili combinazioni; la situazione è mostrata in **figura 3**, relativa a un visualizzatore con 4 digit FND500, ciascuno controllato da un 9368, nell'ipotesi che i codici BCD in ingresso ai 4 Decoder siano quelli corrispondenti alle cifre 0236; con questa predisposizione (ingresso RBI del Decoder collegato al digit più significativo forzato a massa) lo "0" corrispondente al codice $(0000)_2$, fornito sui suoi ingressi DCBA, non viene mostrato (il digit rimane spento).

La terza parte di questo componente è decisamente interessante ed esclusiva: ciascuna delle uscite interne del Decoder è dotata potente Driver in grado di erogare verso i segmenti-LED una corrente $I_{OH}$ da 16 mA (minima) a 22 mA (massima), senza bisogno di resistori per limitare la corrente; la tensione nominale d'uscita a livello alto, $V_{OH}$, è 3,4 V (secondo il datasheet) ma, con un LED a essa direttamente collegato, si adatterà al valore della caduta ai suoi capi, cioè da 1,75 V a 1,85 V.

La **figura 4a** mostra lo schema necessario per pilotare con due 9368 una coppia di digit FND500 (a catodo comune); l'utilizzo della porta PortA del single-chip (o del Registro Dati 0378H della porta parallela di PC) è dedicato alla memorizzazione dei due nibble, uno per unità di visualizzazione: non appena il byte che li raccoglie è stabile sulle sue 8 linee d'uscita interviene il Decoder da Binario puro a 4 bit a "sette segmenti", trasformando immediatamente la codifica binaria in simboli sul digit.

In essa appare evidente che la presenza di resistori di limitazione corrente in serie ai segmenti-LED non è dunque necessaria, sottolineando anche che, nel normale funzionamento, il piedino di controllo LE (pin3) della memoria latch deve

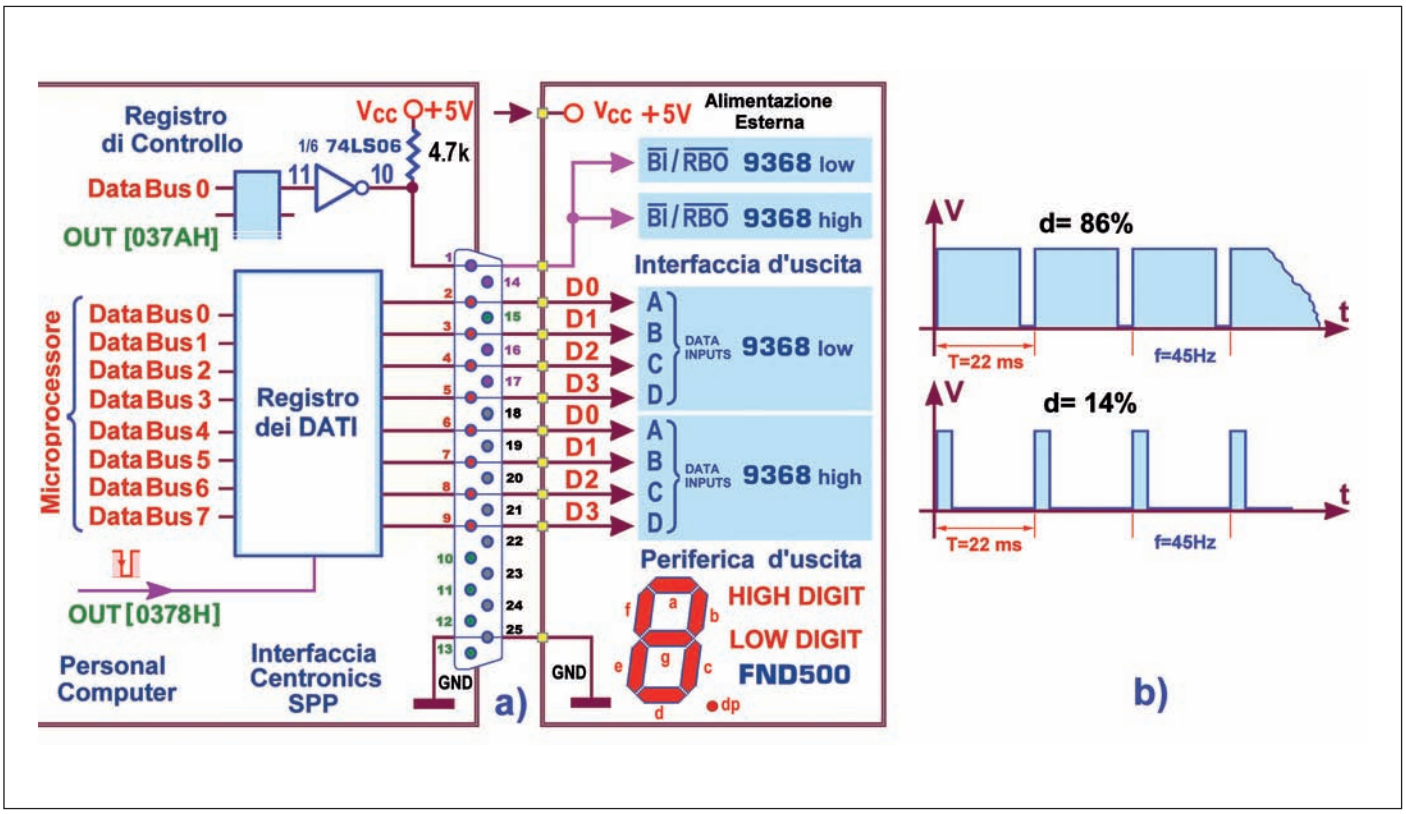



*Figura 5: controllo dell'effetto Blanking su interfaccia d'uscita con due unità 9368/FND500.*

essere a massa mentre le due linee di controllo del Ripple Blanking, RBO (pin4) e RBI (pin5), irrilevanti in questo caso, possono essere lasciate inattive, lasciando scollegata la prima e scollegata (o meglio, collegandola a +5 V) la seconda.
Tuttavia, poiché in virtù di una particolare circuiteria interna (basata su un resistore di pull-up, adatta a realizzare una logica di tipo wired-AND con ogni uscita open collector) la linea del 9368 collegata al pin4 può essere utilizzata, senza produrre danni, sia come uscita (rendendo disponibile il segnale RBO) sia come ingresso (consentendo di forzare su di essa il segnale BI, per spegnere (se posta a 0) il digit collegato al Decoder; nel nostro progetto il valore di quest'ultimo è affidato (facoltativamente) a una delle 8 linee d'uscita di una seconda porta (PortB) del single-chip (oppure ad una linea del Registro di Controllo 037AH della parallela).
La **Figura 4b** mostra la Tabella di Associazione Logica: il suo contenuto propone la configurazione del byte da spedire sulla porta d'uscita; in questo caso ciascuno dei suoi 8 bit è affidato a quelli delle due quaterne da fornire agli ingressi dei 2 9368, distinte in parte alta (High Digit) e parte bassa (Low Digit); il livello attivo da fornire ai segmenti di ciascun digit è ora irrilevante, essendo compito affidato ai Decoder.
La disponibilità di questa struttura virtualizza la presenza dei due digit, offrendo ogni dettaglio utile alla creazione del programma necessario per la loro gestione, cioè senza dover analizzare lo schema elettronico ad essi relativo, anche se la persona chiamata a progettare il codice non è la stessa che ha progettato il circuito.
Il progetto è facilmente adattabile alla porta parallela di un Personal Computer, anche con riferimento al semplice standard SPP: nonostante sia sempre meno presente, essa è facilmente disponibile con l'aggiunta di un'economica scheda di I/O e rimane pur sempre un eccellente

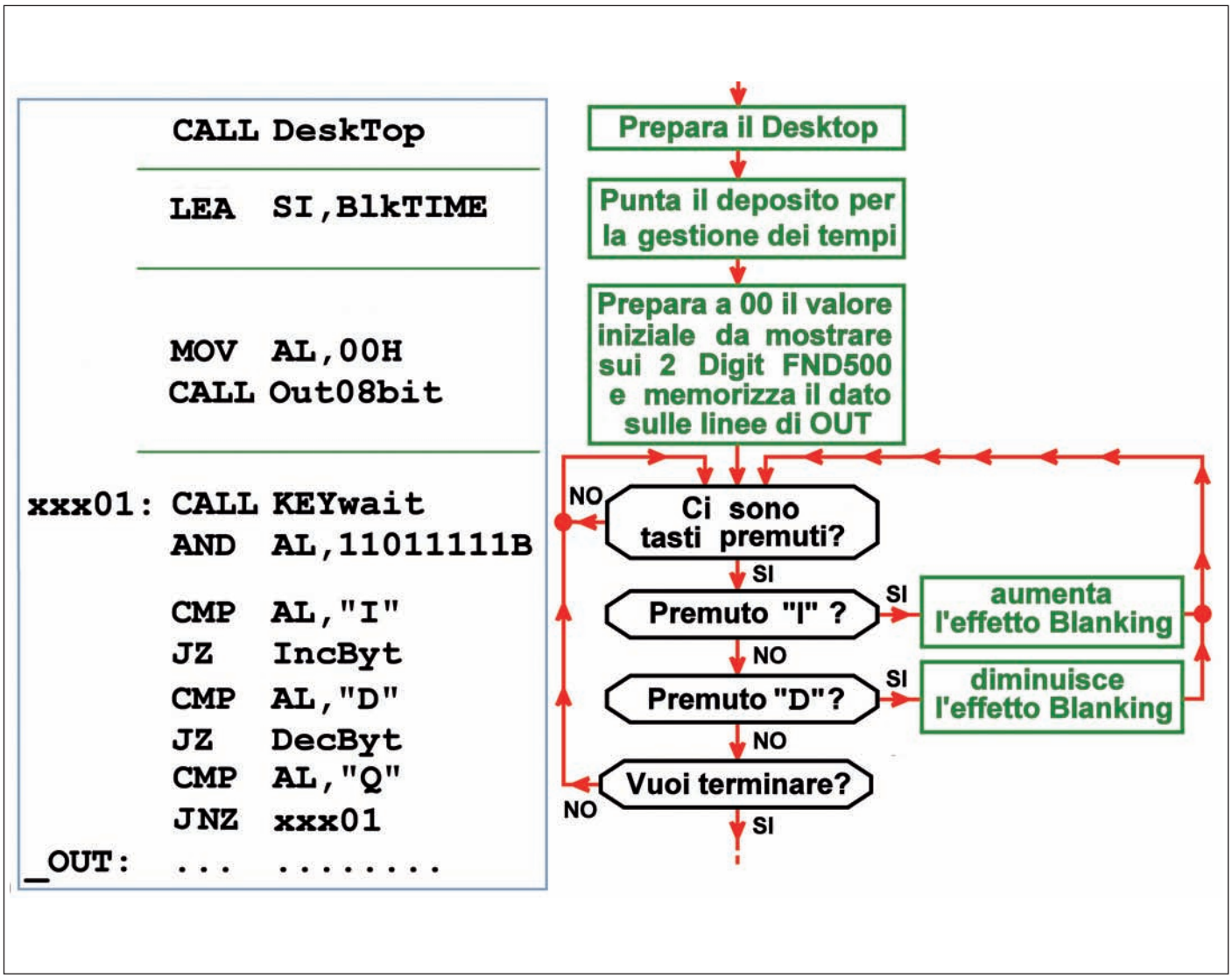



*Figura 6: codice assembly per la gestione dell'effetto Blanking su due unità 9368/FND500.*

strumento per interfacciare numerosi dispositivi esterni con la potenza indiscutibile di un PC, di gran lunga superiore a quella di un single-chip.
Riservandomi di fornire (nel prossimo futuro) un programma in grado di gestire in tutti i modi possibili un conteggio progressivo sui due digit interfacciati dai decoder/latch/driver 9368 vediamo ora il codice necessario per controllare a basso livello l'intensità luminosa del visualizzatore con la tecnica di Blanking (spegnimento): è sufficiente applicare sul pin4 di entrambi gli integrati un'onda quadra non simmetrica e modificare il suo duty cycle (che, come è noto, esprime anche in percentuale il rapporto tra la durata della parte alta e il periodo del segnale) al fine di garantire una sequenza di spegnimenti (imposti da BI attivo, cioè a 0) e accensioni (imposti da BI disattivo, cioè a 1) sufficientemente veloce.
La **figura 5a** mostra lo schema delle parti interne della parallela necessarie per pilotare la parte destra della **figura 4a**, cioè le due unità interfaccia/periferica 9368/FND500.
Se l'aspetto dell'onda quadra è quello di un treno d'impulsi di breve durata, molto distanziati tra loro (tipico di un duty cycle molto basso, per esempio del 5%) l'effetto sui digit è quello di tenerli (quasi) spenti, mentre se l'onda quadra assomiglia ad un treno d'impulsi molto lunghi, praticamente attaccati l'uno all'altro (tipico di un duty cycle molto alto, per esempio del 95%) l'effetto sui digit è quello di accenderli (quasi) al massimo.
La **figura 5b** mostra i segnali necessari rispettivamente per una discreta (duty cycle pari a 86%) e una bassa (duty cycle del 14%) accensione dei digit; l'effetto ottenuto è legato al valore medio del segnale imposto su BI, più vicino a 1 logico nel primo caso e allo 0 nel secondo.
La **figura 6** offre il codice assembly 80x86 necessario per il controllo della luminosità sulle due unità 9368/FND500 via LPT1: dopo la consueta procedura (Desktop) che si occupa della presentazione a video dei compiti del progetto, si provvede a puntare la variabile BlkTIME, una locazione di memoria a 8 bit utilizzata per tener conto del valore corrente della dura-

```
IncByt:  MOV   AL,BlkTIME
         CMP   AL,22
         JZ    All_0
         INC   AL
         JMP   All_0

DecByt:  MOV   AL,BlkTIME
         CMP   AL,1
         JZ    All_0
         DEC   AL

All_0:   MOV  BlkTIME,AL

BlkDgt:
         MOV   CH,00H

         MOV   DX,037AH
         MOV   AL,01
         OUT   DX,AL

         MOV   CL,BlkTIME
Blk_01:  CALL  Del_1
         LOOP  Blk_01

         MOV   AL,00
         OUT   DX,AL

         MOV   AL,BlkTIME
         MOV   CL,23
         SUB   CL,AL
Blk_02:  CALL  Del_1
         LOOP  Blk_02

         MOV   AH,01H
         INT   16H
         JZ    BlkDgt

         JMP   xxx01
```

*Figura 7: codice assembly dei due Servizi necessari per la gestione dell'effetto Blanking.*

ta (in ms) della parte alta dell'onda quadra. Subito dopo, con l'aiuto della procedura Out08bit, il byte 00H viene prima predisposto sul Registro Dati della parallela 0378H e poi trasferito a valle dei due 9368, così da obbligare i due FND500 a mostrare "00"; segue la parte gestionale, sviluppata nell'ambito del Main Loop (cioè è affidata alle istruzioni comprese tra le due etichette xxx01) chiamata a mettere in atto il Servizio relativo al tasto premuto sulla tastiera del PC.
I tasti autorizzati sono solo tre: con "I" (Inc) e con "D" (Dec) si controlla l'incremento o il decremento della luminosità del visualizzatore, mentre con il tasto "Q" (Quit) è possibile terminare il programma e imporre il ritorno immediato al sistema operativo.
La **figura 7** descrive i due Servizi previsti per la gestione di questo progetto: non si tratta di procedure ma di codice specificatamente pensato per rispondere propriamente alla pressione del tasto relativo; a ciascuno di essi si accede infatti con un salto condizionato (dalla flag di Zero, JZ address, la cui presenza a 1 stabilisce che l'evento si è verificato e va servito) e vi si esce con un salto incondizionato (JMP xxx01) direttamente in ingresso al loop d'attesa (non esiste dunque RET).
Ciascuno dei due Servizi (IncByt e DecByt) si limita semplicemente a incrementare o decrementare il valore trovato nella variabile BlkTIME, usato per stabilire contemporaneamente sia la durata t1 della parte alta sia quella t2 della parte bassa della forma d'onda quadra asimmetrica necessaria per il controllo della luminosità; i due tempi sono complementari tra loro e la loro somma t1+t2 è pari al periodo T, volutamente mantenuto (in ogni caso, come si può vedere in **figura 5b**) pari a 22 ms, corrispondente a una frequenza di 45 Hz, la minima in grado di essere percepita (con digit a LED Rossi, all'Arseniuro Fosfuro di Gallio, GaAsP) come variazione continua di luminosità, per effetto della persistenza ottica dell'immagine sulla retina dall'occhio umano. Con digit di altro colore la frequenza è leggermente maggiore, ma la sostanza dell'esperimento non cambia.
L'effettiva onda quadra TTL (cioè con valori tra 0,2 V e 2,4 V) viene generata forzando prima a "1" (per il tempo t1) e poi a "0" (per il tempo t2) il bit0 del registro di Controllo 037AH, per assicurare sul pin1 del connettore della parallela (in virtù del fatto che il valore del bit è invertito internamente) rispettivamente prima un livello basso e poi un livello alto; questa sequenza, offerta all'ingresso BI/RBO (pin4) del "Decoder con memoria" 9368, produce lo spegnimento di tutti i segmenti dei Digit per un tempo t1 e la loro successiva riaccensione per un tempo t2.
Da notare che le temporizzazioni sono rese possibili dalla procedura Del_1, in grado di generare un ritardo fisso di 1 millisecondo con particolare precisione, appoggiandosi per questo al Timer di sistema del personal computer; spetta poi ai due loop interni (Blk_01 e Blk_02) richiamarla il numero di volte necessario per dare all'onda quadra il duty cycle desiderato.
La creazione di ogni periodo T=t1+t2 deve continuare senza interruzioni, per garantire l'effetto Blanking in modo continuativo; per questo è necessario appoggiarsi alla Funzione 01H della procedura a livello BIOS nota come INT 16H, che consente di verificare la eventuale pressione di tasti senza fermarsi; non appena viene rilevata la pressione di un tasto qualunque viene infatti imposto il rientro nel Main Loop, per cambiare comando o per terminare il nostro test.
La versione completa del codice Assembly (Blnk9368.ASM, insieme a numerose note esplicative e all'eseguibile, pronto per l'uso, ottenuto dalla sua compilazione) è disponibile puntando il link di approfondimento consigliato; nella prossima puntata affronteremo (come annunciato) la programmazione di questa struttura 9368/-FND500 per gestire tutti i possibili conteggi, in avanti o indietro. ❑

***CODICE MIP 2827075***

**PER** *approfondire...*

- http://www.giobe2000.it/FareElettronica14.htm

assodel

*Helps you to sell!*
Associazione nazionale fornitori elettronica

FORTRONIC international

# LUMEN ILLUMINOTRONICA
ELETTRONICA PER IL LIGHTING
*...e non solo*

CODICE MIP 2817645

## per chi cerca la **luce**...

## ...cogli al volo le opportunità del **nuovo!**

ASSOCIAZIONI & PARTNER:

IDEA — Federazione internazionale delle associazioni di distribuzione elettronica

VENETO TECHNOLOGIES — Consorzio Ventech

ELETTROMPEX — Consorzio per l'export per le Pmi elettroniche

GRAZIE A:

CNA ARTIGIANCASSA

A CURA DI:

TECNO IMPRESE

**Padova** *presso* Fiera di Padova

**11/13 Ottobre 2012**

SEGRETERIA ORGANIZZATIVA: tel. +39 02 210 111 236
e-mail: *marketing@fortronicforum.com*

**www.illuminotronica.it**

ASSOCIAZIONI & PARTNER:

Comitato Elettrotecnico Italiano

ENEA — Ente per le Nuove tecnologie, l'Energia e l'Ambiente

LUCE IN VENETO — Luci in Veneto

Politecnico di Milano Dipartimento Indaco

IN CONCOMITANZA CON:

casa misura

# Da analogico A DIGITALE

Tutorial PIC18
*(parte quarta)*

**Nel mondo dell'elettronica si verifica spesso la necessità di elaborare segnali che sono di natura analogica come, per esempio, quelli di uscita di un sensore.**
**I PIC dispongono di convertitori AD integrati, utili per questi scopi.**

In questa lezione vedremo nel dettaglio come è gestito il convertitore analogico–digitale di cui sono dotati i PIC, i relativi registri necessari per il funzionamento e le funzioni che il C18 mette a disposizione per la gestione di questa periferica. Per studiare queste novità, al circuito andranno aggiunti due trimmer da utilizzare per simulare le variazioni di tensione di un sensore.

## IL CIRCUITO

Lo schema in questa lezione prevede un leggero cambiamento, rispetto alla lezione precedente, che consiste nello spostare i diodi LED per collegarli a partire dalla porta RA2 fino alla RA5. Questa modifica al circuito permette di utilizzare le due porte RA0 e RA1 come ingressi analogici. Come si può vedere dallo schema, è utilizzata una resistenza da 1 kohm in serie con la resistenza di un trimmer da 5 kohm, mentre il cursore di questo è collegato all'ingresso del pin RA0 del PIC, formando un partitore resistivo su cui si può agire per variare la resistenza da cui prelevare la tensione di uscita che sarà letta dal convertitore AD.
La resistenza da 1 kohm permette di mantenere la configurazione di partitore di tensione anche quando il cursore del trimmer è spostato tutto verso Vcc, quindi resistenza nulla. Utilizzando questa conformazione del circuito possiamo calcolare la tensione massima che avremo all'ingresso dell'ADC con la formula che segue:

```
Vadcmax = Vcc(R2/(R1+R2)) = 4,167
Volt
```

La tensione minima invece arriva a zero volt. Lo stesso discorso vale per la parte di circuito collegata alla porta RA1.

## CONFIGURARE IL CONVERTITORE AD DEL PIC

Molti modelli di PIC includono internamente uno o più convertitori AD. Il PIC 18F2620 usato in questo tutorial, ha un solo convertitore AD con una risoluzione di 10 bit, anche se, leggendo le specifiche nel Data Sheet, potete notare che sono riportati 10 canali analogici (13 per il 18F4620). Questo vuol dire che gli ingressi "analogici" sono internamente collegati a un multiplexer che permette di volta in volta la selezione, tramite un apposito registro, del pin di ingresso da collegare al convertitore. Ne consegue che potranno essere utilizzati solamente uno per volta. Va detto che di questi ingressi, utilizzabili come analogici, due possono essere usati per le tensioni di riferimento del convertitore. Anche in questo caso tramite registri di configurazione è possibile stabilire se utilizzare questi pin per i riferimenti o utilizzare semplicemente le tensioni di alimentazione.
Utilizzare riferimenti esterni può essere utile se si devono misurare range di tensione più ristretti, aumentando di conseguenza la risoluzione della misurazione, oppure se si vuole utilizzare una tensione di riferimento molto stabile per avere misure più accurate.
I registri necessari per il funzionamento del convertitore AD sono ADCON0, ADCON1 e ADCON2, mentre i due registri ADRESH e ADRESL sono utilizzati dal PIC per riportare il risultato della conversione. Per

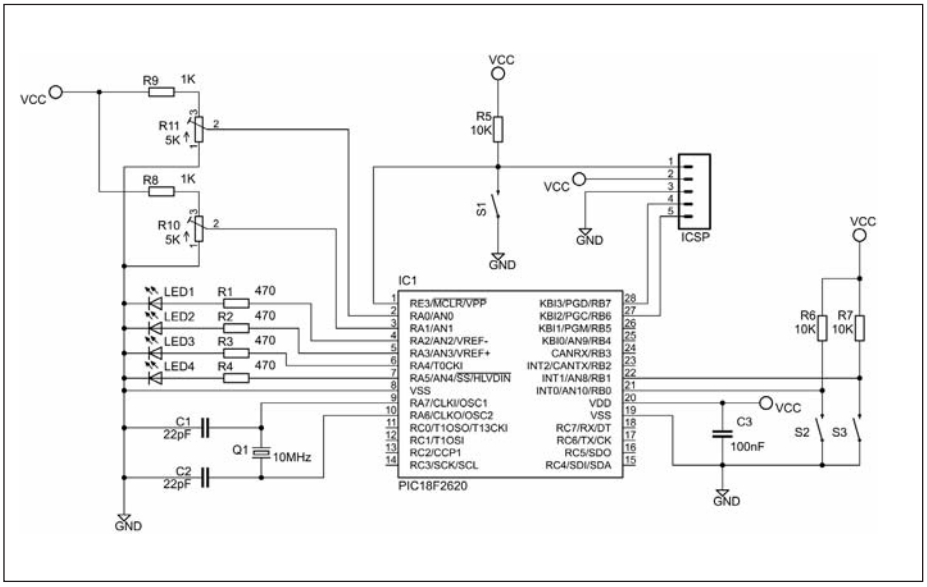



*Figura 1: lo schema del circuito.*

TABELLA 1

| ADCON1 | | | | | | | |
|---|---|---|---|---|---|---|---|
| - | - | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

*Tabella 1: registro ADCON1 e valori impostati.*

prima cosa è bene ricordarsi di impostare i pin che utilizzeremo per l'AD come ingressi agendo sul registro TRISA; il passo successivo è la configurazione del registro ADCON1. Avevamo già visto l'uso di questo registro nelle precedenti lezioni per impostare tutte le porte per funzionare come periferiche digitali, questa volta servirà per impostare le due porte RA0 e RA1 come ingressi analogici. Per farlo sono usati i primi quattro bit di questo registro (PCFG3, PCFG2, PCFG1 e PCFG0), come riportato nella specifica tabella riportata sul Data Sheet del PIC. Nel circuito di questa lezione solo RA0 e RA1 devono essere porte analogiche, le altre saranno tutte digitali quindi, stando alla tabella, si deve inserire il valore 0b1101. I bit quattro e cinque di questo registro servono per impostare le tensioni di riferimento. Nel nostro esempio non abbiamo particolari esigenze, quindi possiamo valorizzare questi bit a 0 per impostare il PIC a utilizzare come riferimenti le tensioni di alimentazione. I restanti bit, non sono utilizzati, quindi il valore finale per la configurazione del registro ADCON1 è 0b00001101.

TABELLA 2

| ADCON2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| ADFM | - | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

*Tabella 2: registro ADCON2 e valori impostati.*

Il registro ADCON2 serve per impostare i tempi che scandiscono la conversione, alcuni dei quali dipendono dalla frequenza di utilizzo del PIC. I primi tre bit servono per impostare il numero di cicli necessari per effettuare la conversione. Questo valore si può ricavare facilmente dal Data Sheet, dove sono riportati – per varie frequenze di utilizzo del PIC – i relativi valori da utilizzare. Nel nostro caso con una frequenza di 40 MHz usiamo il valore 0b010 per avere una conversione della durata di 32 Tosc, ovvero 32 periodi dell'oscillatore.
I tre bit successivi servono invece a indicare il tempo a disposizione del circuito Sample/Hold interno al PIC per avere un valore di tensione stabile per il convertitore. Questo valore va calcolato sulla base di alcune formule riportate sul Data Sheet; nel caso dell'esempio abbiamo livelli di tensione con variazioni molto lente, quindi possiamo fare anche a meno di questo circuito impostando tutti e tre i bit a zero.
Il bit sei non è utilizzato, mentre il settimo serve a indicare l'allineamento del risultato della conversione nei registri ADRESH e ADRESL. Come già visto il convertitore AD restituisce un valore a 10 bit che non può essere contenuto in un normale registro a 8 bit, con la conseguenza che è necessario utilizzare due registri. Per questo motivo si può scegliere come allineare il risultato. Utilizzando l'allineamento a destra avremo che il registro ADRESL conterrà i primi 8 bit del risultato e ADRESH gli ultimi due. Quindi, per concludere, il registro ADCON2 sarà valorizzato con il valore 0b10000010.
Vediamo per ultimo il registro ADCON0, che è quello che sarà utilizzato per abilitare il convertitore AD tramite il bit 0 (ADON), per selezionare il pin su cui fare la conversione tramite i bit dal 5 al 2 (CHS3-CHS0) e per comandare l'avvio della conversione tramite il bit 1 (GO/DONE), il quale sarà utilizzato anche per verificare la conclusione della stessa, in quanto, una volta terminata, tornerà automaticamente al valore 0.

## UN ESEMPIO PRATICO CON IL CONVERTITORE

Vediamo ora il codice del **listato 1** che, tramite il convertitore AD legge il valore di tensione presente sul cursore del trimmer R11, accendendo quanti più LED quanto più alto è il valore di tensione.
Per semplicità di lettura, il codice non riporta la definizione di tutti i pragma, che sono comunque necessari per il funzionamento dello stesso. Le prime novità che potete notare, rispetto agli esempi delle lezioni precedenti, riguardano le define dei LED che sono stati spostati fisicamente su altri pin della PORTA. Poi c'è l'impostazione di tutti i registri ADCONx, così come è riportato nel paragrafo precedente e, in ultimo, la configurazione del registro TRISA, per definire le due porte RA0 e RA1 come input.
La conversione è avviata agendo sul secondo bit del registro ADCON0. Per farlo viene utilizzata la seguente istruzione, che ci permette di agire sullo specifico bit del registro ADCON0:

```
ADCON0bits.GO=1;
```

A questo punto si deve attendere la fine della conversione. Come abbiamo visto in precedenza uno dei modi per determinare la fine dell'operazione è di controllare continuamente lo stato di questo bit. Per farlo sappiamo, dalla lezione precedente, che possiamo utilizzare l'istruzione while. La condizione si baserà sul valore del bit di controllo di fine conversione che passerà a zero solo quando questa sarà terminata, determinando di conseguenza anche la fine del loop e la ripresa del programma.

```
while(ADCON0bits.GO);
```

Nell'include del PIC 18F2620 è definito un oggetto che contiene già concatenati i valori dei due registri ADRESH e ADRESL, il cui nome è ADRES. Nel codice, il valore restituito è utilizzato per valorizzare una variabile che sarà poi utilizzata per determinare quanti LED si devono accendere.
Una piccola nota sul valore letto dall'ADC: sappiamo che il convertitore AD del PIC ha una risoluzione di 10 bit, il che equivale a dire avere $2^{10} = 1024$ possibili valori; inoltre sappiamo che la tensione di riferimento del convertitore è quella di alimentazione che corrisponde a 5 volt; quindi se dividiamo il valore massimo di tensione per 1024 otteniamo la variazione minima di tensione che è possibile leggere tramite il convertitore, corri-

TABELLA 3

| ADCON0 | | | | | | | |
|---|---|---|---|---|---|---|---|
| - | - | CHS3 | CHS2 | CHS1 | CHS0 | GO/DONE | ADON |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

*Tabella 3: registro ADCON0 e valori impostati.*

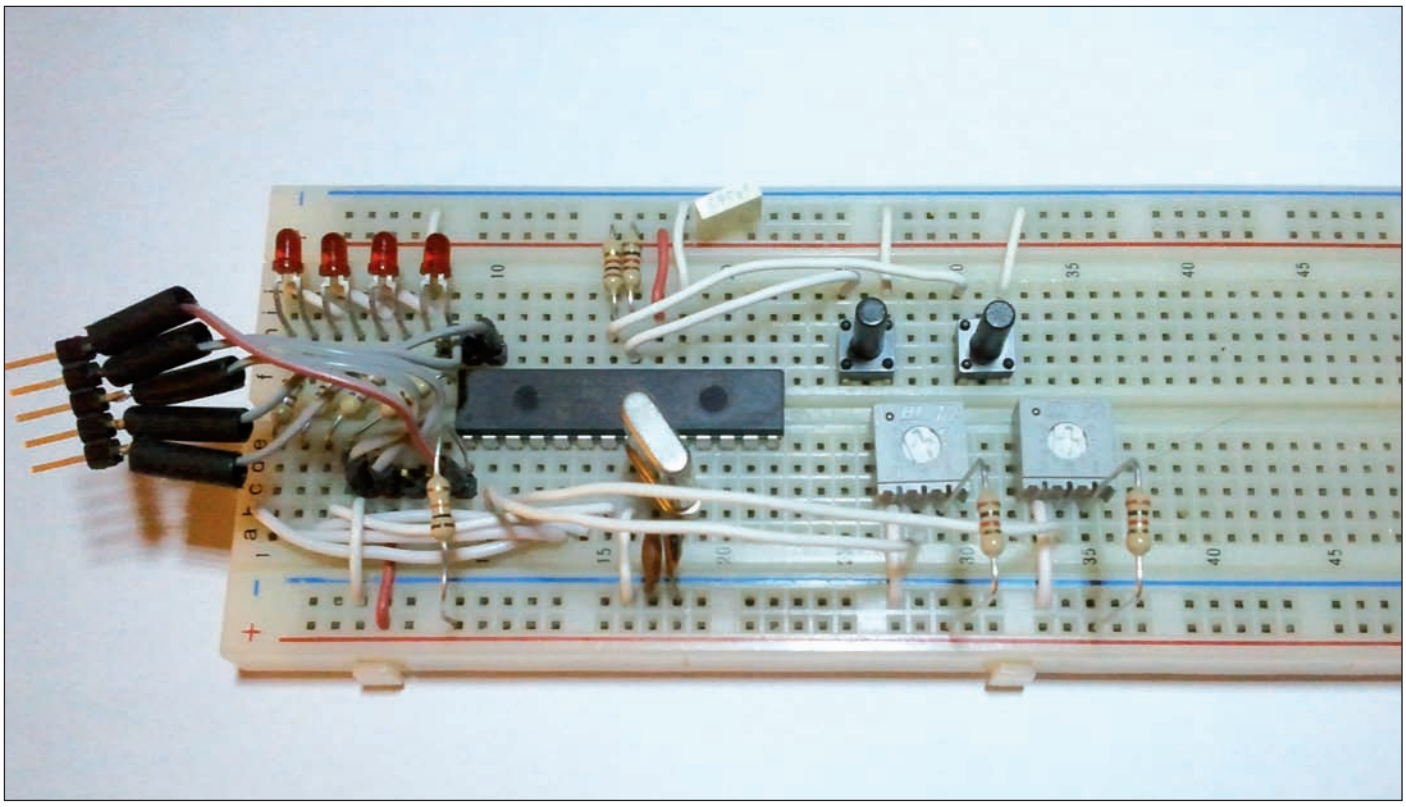


*Figura 2: il circuito montato sulla bread-board.*

# entra in un mondo di contenuti digitali

CLUB DELLE RIVISTE

E-BOOK

ANNATE IN PDF

fare elettronica DIGITAL EXTRA

CODICE MIP 2827215

il riferimento per tutta l'elettronica sul tuo PC o TABLET

**E-BOOK**

I migliori libri di elettronica in formato e-book.
Puoi leggere e stampare i capitoli o le pagine di interesse (ad alta risoluzione) e installare i software eventualmente trattati nei volumi.

**ANNATE IN PDF**

Le annate complete delle riviste Fare Elettronica, Firmware ed Elektor in comodi file zip navigabili o masterizzabili su CD o DVD completi di tutte le riviste (e relativi downloads) uscite nei vari anni, da non perdere!

**CLUB DELLE RIVISTE**

Scarica illimitatamente per un anno centinaia di articoli pubblicati sulle riviste Fare Elettronica e Firmware.

**fare elettronica DIGITAL EXTRA**

Monografie di elettronica in formato digitale: raccolte di progetti completi, CNC e microcontrollori.

www.ieshop.it

INWARE EDIZIONI

LISTATO **1**

```
#include <p18f2620.h>
#include <delays.h>
/**********************
* DEFINIZIONE PRAGMA *
**********************/
...
/**********
* DEFINE *
**********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
* MAIN *
********/
void main(void)
{
   unsigned int result;
   /* Impostazione dei registri di configu-
razione
      del PIC per abilitare ADC sui pin RA0 e
RA1 */
   ADCON1=0b00001101;
   ADCON2=0b10000010;
   ADCON0=0b00000001;
   CMCON=0b00000111;
   // Reset iniziale delle porte
   PORTA=0;
   PORTB=0;
   PORTC=0;
   // Definizione della funzione delle singole
porte
   TRISA=0x03;
   /* I primi due bit della PORTB sono con-
figurati
       come ingressi a cui sono collegati i
pulsanti */
   TRISB=0b00000011;
   TRISC=0x00;

   // Accende il LED4 per indicare che il pro-
gramma è avviato
   LED1=1;
   LED2=1;
   LED3=1;
   LED4=1;
   Delay10KTCYx(250);
   LED1=0;
   LED2=0;
   LED3=0;
   LED4=0;
   // Loop infinito
   for(;;)
   {
       ADCON0bits.GO=1;
       while(ADCON0bits.GO);
       result = ADRES;

       if(result>0)
       {
           LED4=1;
       }
       else
       {
           LED4=0;
       }

       if(result>200)
       {
           LED3=1;
       }
       else
       {
           LED3=0;
       }

       if(result>400)
       {
           LED2=1;
       }
       else
       {
           LED2=0;
       }

       if(result>600)
       {
           LED1=1;
       }
       else
       {
           LED1=0;
       }
   }
}
```

spondente a 0,00488 volt. Come abbiamo visto nella descrizione dello schema, possiamo avere in ingresso una tensione che varia tra 4,167 e 0 volt. Dividendo il valore massimo di tensione che il PIC avrà in ingresso per la variazione minima che è in grado di percepire il convertitore otteniamo che per 4,167 volt abbiamo, come valore letto dall'AD, 853.

Naturalmente questi sono solo numeri approssimativi, nella realtà si devono prendere in considerazione le tolleranze dei vari componenti e gli eventuali errori introdotti dal convertitore stesso.

Sulla base dei calcoli appena fatti, si possono comprendere meglio i valori delle soglie utilizzate per i controlli if presenti nel programma. Come si può vedere, il LED4 si accende solo se il valore è maggiore di 0, il LED3 solo se maggiore di 200, e così via anche per gli altri due.

Possiamo verificare che ruotando il trimmer da un estremo all'altro, i LED an-

## LISTATO 2

```
#define USE_OR_MASKS
#include <p18f2620.h>
#include <delays.h>
#include <adc.h>
/**********************
* DEFINIZIONE PRAGMA *
**********************/
...
/**********
* DEFINE *
**********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
* MAIN *
********/
void main(void)
{
   unsigned int result;
    unsigned char ADConfigReg, ADConfigReg2,
PortConfig;
   /* Impostazione dei registri di configu-
razione
     del PIC per abilitare ADC sui pin RA0 e
RA1 */
   ADConfigReg=ADC_FOSC_32 | ADC_RIGHT_JUST |
ADC_0_TAD;
     ADConfigReg2=ADC_CH0  |  ADC_INT_OFF  |
ADC_REF_VDD_VSS ;
   PortConfig=ADC_2ANA;
   OpenADC(ADConfigReg,ADConfigReg2,PortCon-
fig);
   // Reset iniziale delle porte
   PORTA=0;
   PORTB=0;
   PORTC=0;
   // Definizione della funzione delle singole
porte
   TRISA=0x03;
    /* I primi due bit della PORTB sono con-
figurati
       come ingressi a cui sono collegati i
pulsanti */
   TRISB=0b00000011;
   TRISC=0x00;

   // Accende il LED4 per indicare che il pro-
gramma è avviato
   LED1=1;
   LED2=1;
   LED3=1;
   LED4=1;
   Delay10KTCYx(250);
   LED1=0;
   LED2=0;
   LED3=0;
   LED4=0;
   // Loop infinito
   for(;;)
   {
       ConvertADC();
       while(BusyADC());
       result=ReadADC();

       if(result>0)
       {
           LED4=1;
       }
       else
       {
           LED4=0;
       }

       if(result>200)
       {
           LED3=1;
       }
       else
       {
           LED3=0;
       }

       if(result>400)
       {
           LED2=1;
       }
       else
       {
           LED2=0;
       }

       if(result>600)
       {
           LED1=1;
       }
       else
       {
           LED1=0;
       }
   }
}
```

dranno ad accendersi o spengersi uno alla volta a seconda del valore della tensione in ingresso al convertitore AD in quel momento.

## LE FUNZIONI DEL C18 PER GESTIRE L'ADC

Il C18 fornisce alcune librerie per la gestione delle periferiche, tra le quali troviamo anche quelle per la gestione del Convertitore AD. Per utilizzarle è sufficiente aggiungere tra gli include del programma il file adc.h, come riportato nel **listato 2**.
Analizzando le differenze tra i due listati, si può notare nelle prime righe che, oltre ad aver aggiunto la direttiva include per il file adc.h, è stata aggiunta una define alla prima riga. Questa è necessaria per il corretto funzionamento della procedura di configurazione del PIC. In pratica, inserendo nel codice #define USE_OR_MASKS, è indicato al compilatore quali sezioni del file adc.h devono essere utilizzate per il programma. Tutte queste informazioni sono disponibili nella documentazione del C18, dove per ogni PIC potete trovare le indicazioni delle funzioni per ogni periferica di cui dispone, con tanto di esempi di codice che spiegano come vanno utilizzate.
Potete provare ad aprire il file adc.h, senza modificarlo, e noterete la presenza di controlli sul tipo di PIC che si sta utilizzando, questo perché la libreria deve essere compatibile con tutti i PIC della serie 18.
Vediamo ora come si procede alla configurazione dei registri utilizzando la libreria. Per farlo si utilizza la funzione OpenADC, la quale richiede come parametri i valori da utilizzare per le impostazioni. Un ulteriore supporto dalla libreria viene da una serie di etichette che rendono il codice più leggibile. Nel **listato 2** utilizziamo tre variabili ADConfigReg, ADConfigReg2 e PortConfig, in cui inserire i valori che saranno passati come parametri alla funzione OpenADC.
Vediamo nel dettaglio come è valorizzata la prima variabile. In pratica vanno indicate le stesse impostazioni che abbiamo configurato per il registro ADCON2 nel **listato 1**, con la differenza che, grazie alle define della libreria, possiamo scrivere un codice molto più semplice da leggere, mettendole in OR tra di loro per ottenere il valore da impostare sul registro del PIC. L'etichetta USE_OR_MASK, di cui abbiamo parlato prima, serve proprio per utilizzare le define della libreria con i valori corretti per essere aggregati utilizzando l'operatore OR, come riportato di seguito.

```
ADConfigReg = ADC_FOSC_32 |
ADC_RIGHT_JUST | ADC_0_TAD;
```

Abbiamo detto che ADC_FOSC_32, ADC_RIGHT_JUST e ADC_0_TAD rappresentano le etichette di alcune define presenti nella libreria adc.h. Controllando nella libreria possiamo risalire al valore corrispondente:

```
#define ADC_FOSC_32 0b00000010
#define ADC_RIGHT_JUST 0b10000000
#define ADC_0_TAD 0b00000000
```

Applicando tra di loro l'operatore OR, il che vuol dire che per ogni byte è eseguito un OR logico, bit per bit di tali valori, si ottiene il seguente risultato:

```
0b00000010 | 0b10000000 |
0b00000000 = 0b10000010
```

Se confrontiamo questo risultato con il valore che abbiamo utilizzato nel **listato 1** per configurare il registro ADCON2, possiamo notare che essi coincidono. Ora che abbiamo capito come funziona la logica di questa libreria, è facile interpretare che cosa configurano le altre due variabili che sono passate come parametri alla funzione OpenADC.
Tornando al programma, possiamo vedere che per avviare la conversione è sufficiente chiamare la funzione ConvertADC, la quale non farà altro che assegnare a 1 l'apposito bit del registro ADCON0. Uno dei vantaggi dell'utilizzo delle librerie è che non è richiesta una specifica conoscenza del funzionamento del PIC, ma basta sapere quali funzioni devono essere richiamate per scrivere un programma funzionante.
Per controllare la fine della conversione si utilizza la funzione BusyADC, che ritorna zero solo quando l'operazione è conclusa, quindi si utilizza anche in questo caso l'istruzione while. Per finire, tramite l'istruzione ReadADC si può recuperare il valore misurato dall'ADC. Il resto del programma, che gestisce la logica con cui sono accesi i LED, è uguale a quello del **listato 1**.

## LA PUNTATA SCORSA

Se avete provato gli esperimenti suggeriti nella precedente lezione, avrete notato che aggiungendo un ritardo nel ciclo for, i LED di segnalazione tendono ad accendersi e spegnersi in ritardo rispetto al momento in cui è premuto o rilasciato il pulsante. Questo avviene perché, prima che sia effettuato il controllo sul pin di ingresso da parte del codice, il PIC è impegnato in altre attività, in questo caso provocando un ritardo. Vedremo in seguito come si può gestire un intervento immediato in caso di eventi come il cambio di stato di un pin.
Aggiungendo un ritardo prima del ciclo while nel listato 3 si nota che normalmente lo scorrimento dei LED è più lento, e anche in questo caso quando si preme il pulsante, l'accensione di tutti i LED è in ritardo rispetto al momento in cui esso è premuto.
Modificando il codice per utilizzare l'istruzione do...while, il programma non funziona più correttamente. Questo perché, come abbiamo visto, il codice contenuto all'interno del loop è eseguito almeno una volta, determinando per ogni iterazione del ciclo l'azzeramento della variabile counter utilizzata per il controllo della rotazione dei LED.

## CONCLUSIONI

Come ulteriore verifica e approfondimento di quanto abbiamo trattato fino ad ora, provate a scrivere un programma che legge i valori di tensione su entrambi gli ADC e accende più LED quando questi due valori sono vicini tra loro. ❑

**CODICE MIP 2821441**

# convertitori USB-seriale

## compatti, performanti e convenienti

CODICE MIP 2827219

SystemBase

Per maggiori informazioni visita il sito **www.elettroshop.com** oppure chiama il numero **02/66504794**

www.exposave.com

Fiera Verona
24-25 ottobre 2012

# SAVE

Sponsored by

VERONAFIERE
EIOMI
RITTAL
NATIONAL INSTRUMENTS

Mostra Convegno delle Soluzioni e Applicazioni Verticali di Automazione, Strumentazione, Sensori.

## L'evento verticale di riferimento
## Tecnologie, Soluzioni Applicate, Formazione

- Sistemi di Controllo (DCS, PLC, PC Industriali)
- Strumentazione Industriale di Misura e Controllo
- Interfacciamento HMI
- Sensoristica
- Sistemi di Visione
- Quadri e Regolatori
- Software per l'Industria
- SCADA, Telecontrollo e Reti Tecnologiche
- Strumentazione da Laboratorio
- Valvole e Attuatori
- Efficienza Energetica
- Motion Control
- Fieldbus e Comunicazione

Il miglior rapporto qualità - prezzo

- Stand Preallestiti
- Workshop Tecnici
- Incontri Verticali

CODICE MIP 2819454

# makers italy

Don't Be Bored... Do Something!

**9-11 NOVEMBRE 2012**

## LA NUOVA FIERA-LABORATORIO DEDICATA ALL'INVENTIVA E ALL'INTRAPRENDENZA DEI MAKERS E DEI CREATIVI TECNOLOGICI ITALIANI

Con il patrocinio di:

RegioneLombardia
Commercio, Turismo e Servizi

- 3D PRINTING & IMAGING
- APP & DIGITAL
- PIATTAFORME OPEN SOURCE
- COMPUTER HARDWARE & SOFTWARE
- CHIMICA
- DESIGN
- ELETTRONICA ED ELETTROTECNICA
- IDROGENO & CELLE A COMBUSTIBILE
- INGEGNERIA
- LEGO AREA
- MATERIE PLASTICHE
- MECCANICA & MECCATRONICA
- MOBILITÀ SOSTENIBILE & VEICOLI ELETTRICI
- RADIO & HAM RADIO
- ROBOTICA
- SOSTENIBILITÀ & ENERGIE ALTERNATIVE
- VIDEO, AUDIO & MOVIE
- WIRELESS ELECTRICITY / TESLA

CODICE MIP 2826454

**www.makersitaly.it**

a cura della redazione

# Più memoria per i chip Toshiba

*Toshiba aumenta la capacità delle memorie ROM e RAM nei microcontrollori per azionamenti ad alte prestazioni e bassa potenza*

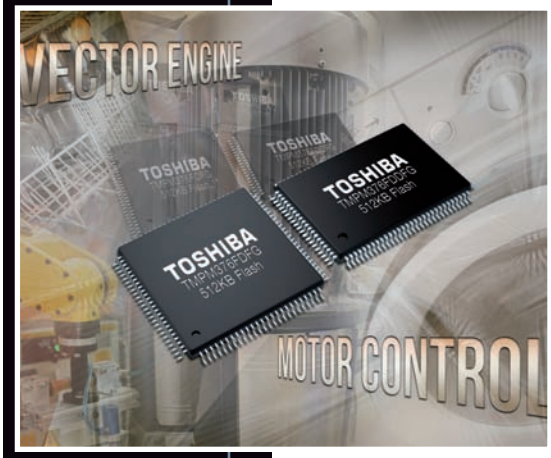
*Toshiba Electronics Europe estende ulteriormente la famiglia di microcontrollori ARM Cortex-M3 a 32 bit per il controllo di motori con un nuovo dispositivo dotato della più capiente memoria integrata attualmente disponibile nella serie. Il chip TMPM376 riduce al minimo il numero dei componenti e velocizza la progettazione degli azionamenti per apparecchiature domestiche e industriali che sfruttano il controllo ad orientamento di campo (Field Oriented Control) per il pilotaggio efficiente e preciso di ventole, pompe e compressori. Il dispositivo TMPM376 di Toshiba, ad alte prestazioni e bassa potenza, ha tutte le caratteristiche necessarie per garantire un controllo preciso del motore e una risposta rapida senza dover aggiungere altro hardware, appesantire il carico generale sulla CPU o perdere del tempo prezioso a sviluppare software. Oltre a fornire le funzioni richieste per un controllo preciso di motori trifase, il dispositivo contiene inoltre al suo interno 512 kB di memoria ROM e 32 kB di memoria RAM, riducendo al minimo la necessità di memoria esterna per codice e dati.*

*Le funzioni di controllo motore del TMPM376 sono svolte da un motore vettoriale (Vector Engine, VE) hardware integrato e da uno stadio di azionamento programmabile PMD3+ a due canali. Il VE effettua i calcoli necessari per controllare fino a due motori mentre il PMD3+, dotato di generatore PWM trifase, controller del tempo morto e circuito di protezione, fornisce i segnali PWM per il controllo del motore. Un ADC a 12 bit supporta una conversione analogico-digitale ad alta velocità, sincronizzata al segnale PWM, mentre un circuito di ingresso encoder a due canali con contatore e comparatore facilita il rilevamento del senso di rotazione e della posizione del motore.*

*Oltre al nucleo ARM, alla memoria e ai circuiti di azionamento, il TMPM376 comprende un timer a 8 canali e 16 bit, un watchdog timer, un'interfaccia seriale universale che offre la possibilità di scegliere tra modalità UART e sincrona e un'interfaccia a bus seriale commutabile tra funzionamento I2C oppure sincrono. Sono disponibili, come caratteristiche di serie, funzioni di ripristino all'accensione (Power on Reset, POR) e di rilevamento della tensione (Voltage Detect, VLTD), mentre la possibilità di rilevare la frequenza di oscillazione (Oscillation Frequency Detect, OFD) permette di rispettare la normativa sulla sicurezza degli elettrodomestici.*

*Disponibile nei contenitori QFP da 14 x 20 mm (TMPM376FDDFG) e LQFP da 14 x 14 mm (TMPM376FDFG), il nuovo microcontrollore funziona con tensioni di alimentazione da 4,5 V a 5,5 V e con una frequenza di lavoro massima di 80 MHz. Un generatore di clock integrato fornisce rapporti di clock di 1/1, 1/2, 1/4, 1/8 e 1/16. Il nucleo ARM a bassa potenza, la memoria NANOFLASH ad alta efficienza e le modalità di standby IDLE e STOP contribuiscono nell'insieme a ridurre il consumo energetico dell'applicazione finale.*

*I chip TMPM376: alte prestazioni e bassa potenza*



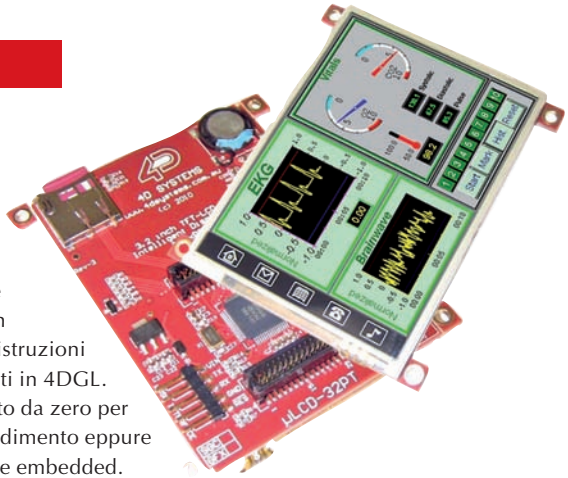
Hai un quesito di elettronica?
Vuoi comunicare con altri appassionati?
Vuoi condividere le tue esperienze?
Tutto questo su IEforum!
CODICE MIP 2827223
www.ieforum.it

# Lo shop dei prodotti Inware Edizioni...

PRIMO PIANO

## Modulo LCD intelligente

uLCD-32PT(GFX) è un modulo LCD "SMART" da 3.2" all-in-one compatto ed economico che fa uso della più recente tecnologia LCD a matrice attiva (TFT) e dispone di un controller grafico PICASO-GFX2 integrato che fornisce funzionalità "stand-alone" a qualsiasi progetto. Il PICASO-GFX2 appartiene a una famiglia di processori che dispone di un motore virtuale soft core altamente ottimizzato, E.V.E. (Extensible Virtual Engine). EVE è un processore virtuale proprietario ad alte prestazioni con un set di istruzioni byte-code estensivo ottimizzato per eseguire programmi compilati in 4DGL. 4DGL (4D Graphics Language) è stato specificatamente sviluppato da zero per l'engine core EVE. È un linguaggio ad alto livello di facile apprendimento eppure abbastanza potente per affrontare tantissime applicazioni grafiche embedded.




***CODICE: ULCD-32-PT PREZZO: 79,86 €***

## Arduino - Making things talk

Costruire prototipi elettronici che interagiscono con il mondo circostante è divertente. Ma quando i dispositivi che avete costruito iniziano a comunicare tra loro, il gioco si fa interessante. Con 22 progetti di facile costruzione, Making Things Talk vi mostra come far comunicare i vostri gadget elettronici con voi stessi o con l'ambiente esterno. È perfetto per chi ha poca capacità tecnica ma tanto interesse. Magari siete degli insegnanti che vogliono mostrare ai propri studenti come monitorare il tempo in diversi luoghi simultaneamente. Oppure scultori in cerca di una soluzione per coreografare sculture meccaniche. Con questo libro, imparerete a formare network di dispositivi intelligenti che condividono dati e rispondono a comandi.

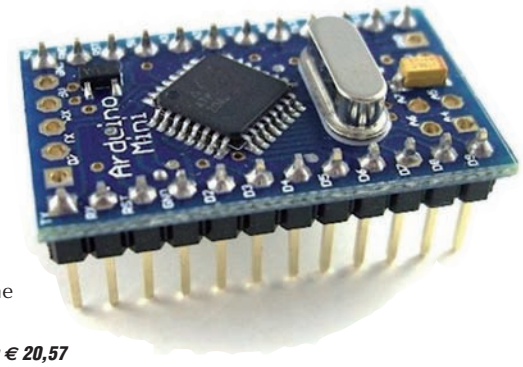
***CODICE: B000002 PREZZO: € 35,09***

NOVITÀ

NOVITÀ

## Arduino Mini

Arduino Mini è una piccola scheda di sviluppo concepita per l'utilizzo su breadboard e in condizioni particolari di spazio. La versione 5 (rev 05) utilizza un altro package dell'ATmega328 che permette a tutti i componenti di essere sulla faccia superiore della scheda, insieme ad un pulsante di reset. Arduino Mini ha 14 pin di input/output (sei dei quali possono essere usati come PWM), 8 ingressi analogici e un oscillatore a 16 Mhz. Può essere programmata con USB Serial adapter o altri adattatori da seriale a USB o RS232 o TTL Serial adpter. Arduino Mini è dotato di bootloader, quindi può essere programmato senza l'ausilio di un programmatore specifico. La versione 05 ha ereditato la stessa disposizione dei pin della 04.

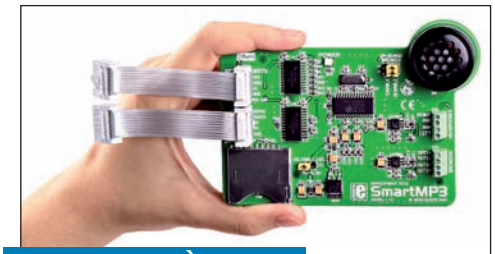



***CODICE: A000087 PREZZO: € 20,57***

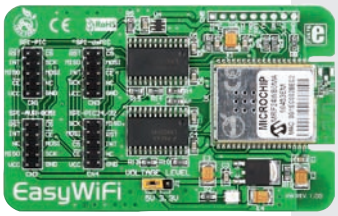
## Crea un lettore MP3

La nuova scheda SmartMP3 è ideale per creare lettori MP3 e l'aggiunta di funzionalità musica e audio per i dispositivi di prototipazione, anche con microcontrollori meno efficienti. La SmartMP3 include, come caratteristiche peculiari, lo slot per MMC/SD card e il decoder MPEG audio single-chip VS1011e. Il chip, ad alte prestazioni, include: DSP processor core VS DSP4 a basso consumo, memoria di lavoro, 5 KiB istruzioni RAM e 0,5 KiB dati RAM per le applicazioni utente, il controllo seriale e dati di interfacce di input, 4 pin I/O general purpose, nonché frequenza di campionamento DAC di alta qualità variabile, oltre ad un amplificatore auricolare e un buffer comune. La scheda può essere facilmente interfacciabile con una scheda di sviluppo fornita di IDC10 utilizzando cavi di connessione.

***CODICE: SMARTMP3 PREZZO: € 35,09***




**NOVITÀ**




**OFFERTA**

## Applicazioni in WiFi

La EasyWiFi Board è uno strumento di sviluppo per dispositivi che utilizzano la comunicazione wireless a 2,4 GHz, capace di offrire un'ampia quantità di features. L'utilizzo ideale di questo strumento è sulle applicazioni embedded. Il modulo on-board ZeroG ZG2100M fornisce una veloce trasmissione dati e un ampio raggio di dati. I protocolli di comunicazione usati per interfacciarsi al proprio sistema o microcontrollore sono gli standard UART e SPI. Inoltre, la board fornisce switch di voltaggio che permettono di lavorare sia su sistemi a voltaggio 3,3 V che 5 V. I connettori IDC10 on-board sono utilizzati per collegare la board a dispositivi elettronici esterni.

***CODICE: EASY-WIFI PREZZO: ~~€ 49,61~~ € 42,35***

## GPS Facile!

Connetti il GPS (Sistema di Posizionamento Globale) al tuo PC o alla tua scheda di sviluppo con semplicità con la board SmartGPS equipaggata con modulo LEA-5S. Il modulo LEA-5S è basato sulla potente tecnologia u-blox 5 50-canali. Un motore di acquisizione a 32-canali con oltre 1 milione di correlatori effettivi è capace di ricerche parallele massive. Ciò permette un Time To First Fix (TTFF) inferiore a 1 secondo e una sensibilità di acquisizione e tracking di -160 dBm. Una funzionalità disponibile è SuperSense® KickStart, una nuova caratteristica che permette l'acquisizione accelerata di segnali deboli. Una volta acquisiti, i satelliti sono passati a un motore di tracking dedicato e ottimizzato. Questa sistemazione permette al GPS e al motore GALILEO di tracciare simultaneamente 16 satelliti durante la ricerca di nuovi altri. L'avanzato meccanismo di soppressione del jamming di u-blox 5 e l'innovativa architettura RF forniscono un'alto livello di immunità al jamming, assicurando le massime prestazioni con GPS e GALILEO.

***CODICE: EP-EASYGPS PREZZO: ~~€ 59,29~~ € 47,19***

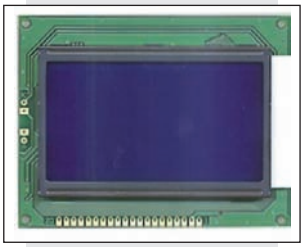
**OFFERTA**

## Modulo grafico LCD 128 × 64

Modulo grafico LCD 128 × 64 dot matrix LCD, STN con retroilluminazione blu. Il controller è compatibile con KS0108.

***CODICE: HG128643BNHDWBD***

***PREZZO: ~~€ 55,96~~ € 42,35***

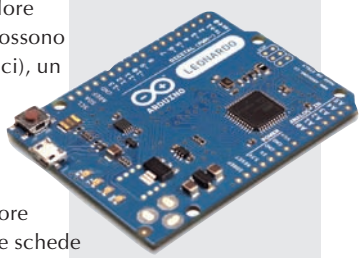


**OFFERTA**

## Arduino Leonardo

Arduino Leonardo è una scheda basata su microcontrollore ATmega32u4. Possiede 20 pin I/O digitali (dei quali 7 possono essere usati come output PWM e 12 come input analogici), un cristallo oscillatore a 16 MHz, una connessione micro USB, un jack per alimentazione, un header ICSP e un pulsante di reset. Contiene tutto il necessario per supportare il microcontrollore; per utilizzarlo, connettetelo a un PC via USB o alimentatelo via adattatore AC>DC o tramite batterie. Leonardo differisce da tutte le schede precedenti in quanto l'ATmega32u4 fornisce comunicazione USB built-in, eliminando il bisogno di avere un processore secondario. Ciò permette a Leonardo di apparire a un computer connesso come un mouse o una tastiera, oltre che ad emulare permanentemente una porta seriale (CDC) / COM port. Questa versione di Leonardo non è fornita di connettori (connettori femmina sui piedini, il connettore ISCP e il connettore di alimentazione).

***CODICE: A000052 PREZZO: € 19,36***




**BEST SELLER**

# Lo shop dei prodotti Inware Edizioni... ma non solo!

## Bundle MikroC

Il compilatore mikroC PRO per i microcontrollori PIC12, PIC16 e PIC18 in Versione Key File License compreso il libro "Pillole di Microcontrollori PIC" (di Paolo Rognoni) Manuale di progettazione con esempi pratici in C ad un prezzo veramente speciale. Con questo bundle avrete tutto l'occorrente per sviluppare applicazioni con i PIC anche molto complesse!

***CODICE: BUNDLE MIKRO-C PREZZO: € 244,09***

**BUNDLE**

## ANNATE Fare Elettroncia 2003-2011

In un comodo file zip da scaricare, tutti i numeri di Fare Elettronica dal 2003 al 2011 in pdf ad alta risoluzione! Potrai così visualizzare le singole riviste e stampare gli articoli che più ti interessano senza perdita di qualità. Nel file zip è incluso anche una comodissima interfaccia utente che ti permetterà di navigare nelle riviste con estrema facilità.

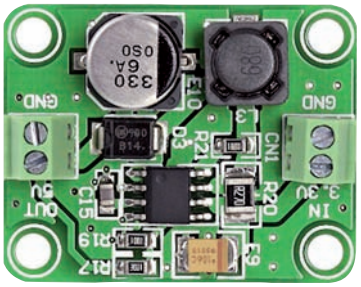
***CODICE: FE-DVD PREZZO: € 20,16***

## Traslatore di livello

Scheda per traslazione di livelli di tensione basata su LVCC3245A. LVCC3245A è prodotto realizzato con l'avanzata tecnologia dual metal CMOS. Questo bus transceiver non-invertente a 8 bit (octal) contiene due rail di alimentazione separate che permettono la transizione da 3,3 V a 5 V. È l'ideale per comunicazione asincrona tra due bus dati (A e B). Trasmette dati da A a B o viceversa, a seconda del livello logico al controllo-direzione (DIR) input. L'output-enable (OE) può essere usato per disabilitare la periferica, in modo da isolare i bus in modo efficace.

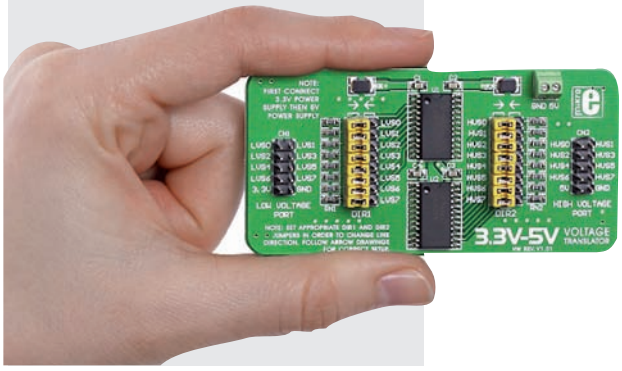
***CODICE: 3.3V-5V VOLTAGE TRANSLATOR BOARD PREZZO: € 22,99***

## DC-DC Converter



Scheda DC/DC basata su MC34063A, un circuito di controllo monolitico contenente le principali funzioni richieste per DC-DC converter. Questi dispositivi sono costituiti da una temperatura interna con riferimento compensato, di confronto, ciclo di funzionamento ad oscillatore controllato da un circuito attivo con limite di corrente, driver e switch ad alta corrente di uscita. Questa scheda è ideale per progetti che richiedono la conversione da 5 V a 3,3 V, come progetti a piccolo microcontroller.

***CODICE: 3.3V-5V VOLTAGE REGULATOR BOARD PREZZO: € 15,73***



## prossimamente su Fare Elettronica

COMING...

### RASPBERRY PI

Continua la rassegna di articoli dedicati a Raspberry Pi con un articolo dedicato alle caratteristiche tecniche di questo dispositivo. Un preludio a una serie di interessantissime applicazioni pratiche.

### CONVERSIONE DA CON I PIC18

Si sa che i microprocessori capiscono solo i numeri 1 e 0, ma spesso si ha la necessità di attivare dei dispositivi che invece prediligono una tensione variabile.

### STAMPANTI 3D

Dopo l'articolo introduttivo sull'affascinante mondo delle stampanti 3D iniziamo ad entrare nei dettagli illustrando gli strumenti hardware e software necessari per la realizzazione degli oggetti.

### COMPUTER VISION

Dopo aver visto che cosa è la Computer Vision e illustrato il processo di installazione delle librerie OpenCV è giunto il momento di fare qualche esperimento pratico. Seguite il tutorial di Walter Lucetti e rimarrete sicuramente sbalorditi!


# COME RISPARMIARE?

**Ordinare direttamente tramite internet conviene! Potrai infatti aggiudicarti dei buoni sconto (coupon) con le modalità:**

**COUPON ABBONAMENTO ONLINE FE E FW**

Sottoscrivendo un nuovo abbonamento o rinnovando il proprio online, direttamente tramite IE SHOP, si ha diritto ad un coupon del 15%! **Qualora ci si abbona (sempre on-line) a Fare Elettronica e Firmware insieme il coupon è del 20%!**

**COUPON QUIZ "ELETTROQUIZ"**

Rispondendo ai Quiz presente ogni mese su Fare Elettronica si avrà diritto ad un coupon del 10%. Tutte le risposte pervenute saranno accuratamente valutate dagli autori dei quesiti e, in caso di risposta esatta (oltre al Coupon) potrai vincere fantastici premi!

**COUPON "ACQUISTI PREMIATI"**

I tuoi acquisti su www.ieshop.it vengono premiati con un coupon del 10% se il tuo ordine supera i 100 EUR (IVA e spese di trasporto escluse). Ma se il tuo ordine supera i 200 EUR, lo sconto sale al 20%!

**IMPORTANTE!**

- Il coupon può essere speso **esclusivamente** ordinando sul sito www.ieshop.it
- Il coupon non potrà essere utilizzato in caso di nuova sottoscrizione o rinnovo dell'abbonamento, è invece utilizzabile per tutti gli altri prodotti presenti su www.ieshop.it
- Ogni coupon ha una scadenza, non dimenticarlo!
- Lo slogan "più acquisti più risparmi" pecca senz'altro di originalità, ma in questo caso è molto azzeccato: i coupon sono infatti utilizzabili solo una volta, usalo bene!
- I coupon non sono cumulabili, e vanno utilizzati singolarmente, secondo il criterio cronologico di assegnazione.



CONTI CORRENTI POSTALI - Ricevuta di Versamento

BancoPosta

sul C/C n. 70107552

di Euro

importo in lettere

INTESTATO A:
INWARE EDIZIONI SRL

ESEGUITO DA:

VIA - PIAZZA

CAP

LOCALITÀ

BOLLO DELL'UFF. POSTALE

AVVERTENZE
Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.
La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.
La ricevuta rilasciata dall'ufficio postale garantisce in modo definitivo l'avvenuto pagamento di quanto in essa riportato.

CAUSALE

CONTI CORRENTI POSTALI - Ricevuta di Accredito

BancoPosta

sul C/C n. 70107552

di Euro

TD 451

importo in lettere

INTESTATO A:
INWARE EDIZIONI SRL

CAUSALE

ESEGUITO DA:

VIA - PIAZZA

CAP

LOCALITÀ

BOLLO DELL'UFF. POSTALE
codice bancoposta

IMPORTANTE: NON SCRIVERE NELLA ZONA SOTTOSTANTE
importo in euro
numero conto

AUT. DB/SISB/E 20017 DEL 11.04.2006

td

70107552 < 451>

# Indagine sui lettori

*Aiutaci*
*a conoscerti meglio!*

*Con il tuo aiuto*
*riusciremo ad offrirti*
*una rivista sempre più*
*in linea con le tue*
*aspettative.*

*Compila online il*
*questionario all'indirizzo*
*www.farelettronica.com/survey*

*Per ringraziarti*
*per il tuo tempo e la tua*
*cortesia, ti invieremo*
*gratuitamente*
*un bellissimo eBook*
*del valore di 14,52 euro!*

**IMPORTANTE: NON SCRIVERE NELLA ZONA SOPRASTANTE!**

AVVERTENZE

Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.

La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.

Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

PRODOTTI

Se vuoi ordinare "offline" ovvero senza usare Internet, usa il seguente modulo d'ordine **PRODOTTI**

| COD. PRODOTTO | DESCRIZIONE | PREZZO UNITARIO | Q.tà | Totale |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | SUB TOTALE | |
| | | | SPESE DI SPEDIZIONE | |
| | | | TOTALE | |

***Nota. Se le righe non sono sufficienti, utilizzare più copie del modulo.***
***Per maggiori dettagli sulle spese di spedizione e i metodi di pagamento vedi retro cartolina.***

**METODI DI PAGAMENTO**

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **CONTRASSEGNO**

FE 327

☐ MasterCard ☐ VISA ☐ postepay

TITOLARE. . . . . .
NUMERO . . . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . . . .

**INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225**
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . Cognome . . . . . .
Indirizzo . . . . . n° . . . . . .
Cap . . . . . Città . . . . . Prov . . . . . .
Tel . . . . . Fax . . . . . Email . . . . . .
Ragione Sociale . . . . . .
Codice fiscale . . . . . Partita IVA . . . . . .

DATA . . . . . FIRMA . . . . . ☐ Fattura

ABBONAMENTO

## *Sì, mi abbono a Fare Elettronica*

☐ Abbonamento TRIAL a 3 numeri di FARE ELETTRONICA a soli € 16,50 anziché € 18,00
☐ Abbonamento PREMIUM a 11 numeri di FARE ELETTRONICA a soli € 55,00 anziché € 66,00
☐ Abbonamento PRO a 11 numeri di FARE ELETTRONICA include il CD dell'annata 2011 a soli € 59,50 anzichè € 96,00

FE 327

**METODI DI PAGAMENTO** (per maggiori dettagli vedi retro cartolina)

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . . .
NUMERO . . . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . . . .

**INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225**
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . Cognome . . . . . .
Indirizzo . . . . . n° . . . . . .
Cap . . . . . Città . . . . . Prov . . . . . .
Tel . . . . . Fax . . . . . Email . . . . . .
Ragione Sociale . . . . . .
Codice fiscale . . . . . Partita IVA . . . . . .

DATA . . . . . FIRMA . . . . . ☐ Fattura

## ISTRUZIONI PER L'ORDINE E SPESE DI SPEDIZIONE

Il modulo d'ordine dovrà essere compilato in tutte le sue parti ed inviato via posta o via fax ai recapiti indicati sul modulo d'ordine stesso. Gli ordini potranno essere fatti anche direttamente online dal sito www.ieshop.it. In questo caso non sarà necessario inviare il modulo d'ordine. Le spese di spedizione ammontano a euro 8,50 a cui vanno aggiunti euro 3,50 se si sceglie di pagare in contrassegno.

## METODI DI PAGAMENTO

Si accettano pagamenti ino in contrassegno, carta di credit, bollettino postale o bonifico all'ordine. Per il contrassegno verrà applicata una spesa aggiuntiva di euro 3,50 per le spese di contrassegno. Forme diverse di pagamento devono essere previamente concordate.

### Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

### Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento.

### Contrassegno

La merce verrà pagata direttamente al corriere alla consegna della merce. Il pagamento in contrassegno comporta l'addebito di euro 3,50 per spese di contrassegno.

### Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

## TERMINI E CONDIZIONI

Tutti i prodotti sono coperti da garanzia. La garanzia copre tutti i difetti di fabbricazione entro un anno dal ricevimento della merce. Tutti i prodotti non funzionanti per uso improprio o incuria non saranno ritenuti in garanzia e saranno addebitati gli eventuali costi di riparazione. Tutti i prodotti verranno riparati e/o sostituiti direttamente dal produttore. Non sono coperti da garanzia i componenti elettronici (microprocessori, memorie, ecc.) La garanzia dei prodotti si intende F.co ns. sede, le eventuali spese di trasporto sono a carico del cliente salvo accordi diversi. Per dar corso alla riparazione/sostituzione in garanzia è necessario seguire l'apposita procedura di RMA.

## PRIVACY

Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni srl, Via Giotto 7 – 20032 Cormano o tramite email all'indirizzo info@inwaredizioni.it

---

## ABBONARSI ALLE RIVISTE INWARE EDIZIONI CONVIENE!

I vantaggi per gli abbonati sono, oltre al prezzo bloccato per un anno, la ricezione del numero direttamente a casa con la garanzia di ricevere tutti i numeri. Inoltre un vistoso risparmio che, nel caso dell'abbonamento PLUS, ammonta a ben 36,50 euro. L'abbonamento ha una durata di 12 mesi e comporta l'invio di 11 numeri di Fare Elettronica. Eventuali variazioni di indirizzo andranno comunicate tempestivamente alla Redazione che provvederà a registrare il cambiamento senza alcuna spesa aggiuntiva. L'abbonamento decorrerà dal primo numero raggiungibile alla data di avvenuto pagamento. Non sono previsti rimborsi in caso di disdetta dell'abbonamento.

## METODI DI PAGAMENTO

### Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

### Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento indicando nella causale: "Abbonamento Fare Elettronica"

### Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

### Assegno bancario

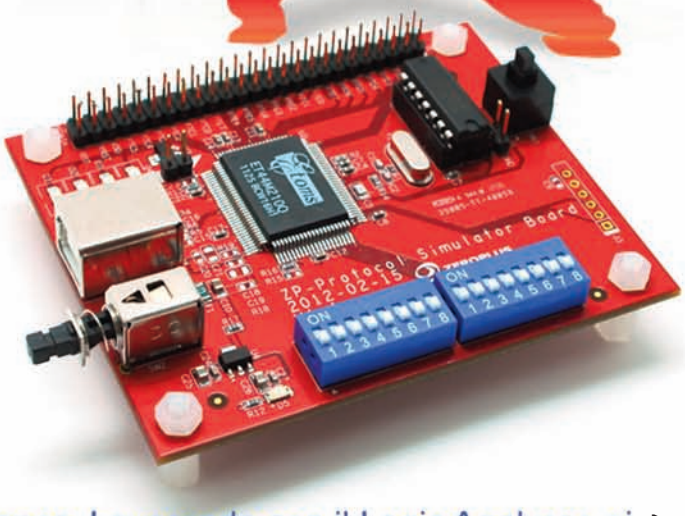
È possibile spedire un assegno bancario insieme a questo coupon.

### Paypal

Inviare il pagamento a paypal@invaredizioni.it.

ZEROPLUS

# Protocol Simulator Board



Problemi per analizzare i segnali BUS presenti nel proprio progetto?
Zeroplus viene in aiuto con questa scheda per l'analisi dei protocolli!
Durante lo sviluppo o la decodifica dei segnali, la scheda "PSB" aiuta a visualizzare i pacchetti e ridurre così i tempi di sviluppo migliorando l'efficienza. Lavorando con il Logic Analyzer, si possono valutare le relazioni della base di tempo e dei pacchetti migliorando il progetto.
Ideale sia per i progettisti professionisti che per la didattica.

CODICE MIP 2827167

| PC System | Memory | Digital Audio | IC Interface | Optoelectronics | Wireless | Other |
|---|---|---|---|---|---|---|
| Compact Flash 4.1 | 1-Wire, 3-Wire | AC97 | HPI | 7-SEGMENT LED | Differential Manchester | BDM, DS 1302 |
| eMMC | I2C | HD Audio | JTAG 2.0 | CCIR656 | DigRF | DS18B20 |
| FWH | I2C(EEPROM 24L) | HDMI CEC | MCU-51 DECODE | COME IMAGE | ISO7816 UART | DSA Interface |
| Low Pin Count | I2C(EEPROM 24L | I2S | MICROWIIRE | DALI Interface | KEELOQ Code Hopping | GPIB, HART |
| LPC-SERIRQ | CS6/24LCS62) | PCM | SIGNIA 6210 | DM114/DM115 | MANCHESTER | ModBus |
| LPT, PCI, PECI | MICROWIRE(EEP | S/PDIF | SLE4442 | DMX512 | MII | MODIFIED SPI |
| PS/2 | ROM 93C) | STBus | SSI Interface | LCD12864 | MIL-STD-1553 | OPENTHERM 2.2 |
| DS2.0/DSIO, SVID | SPI, SPI PLUS | | ST669 | LCD1602 | MILLER | PSB Interface |
| UART(RS232C | SPI(EEPROM AT25F) | | | LG4572 | MODIFIED MILLER | PROFIBUS |
| /422/485) | SAMSUNG | | | S2Cwire/AS2Cwire | SWP | SHT11, YK-5 |
| USB 1.1 | K9(NAND flash) | | | SCCB | WIEGAND | 1-Wire(Advanced) |
| USB 2.0 | UNI/O | | | | WWV/WWVH/WWVB | |

| Infrared rays | Basic Logic Application | Power | Automotive |
|---|---|---|---|
| IRDS | ARITHMETICAL LOGIC | BMS, HDQ | CAN 2.0B |
| NEC PD6122 | DIGITAL LOGIC | PMBus 1.1 | DSI Bus |
| Philips RC-5 | JK FLIP-FLOP | SDQ | FlexRay 2.1A |
| Philips RC-6 | UP DOWN COUNTER | SMBus 2.0 | LIN 2.1 |
| PT2262/PT2272 | | | MVB |
| | | | WTB |

LAP-C Series

LAP-B Series

E' un prodotto distribuito da elettroshop.com brilliant electronics since 1998

ZEROPLUS

Per maggiori informazioni www.elettroshop.com oppure chiamare lo 02/66504794

www.sena.com

ZigBee Member

# ProBee Series

## Il nuovo standard per la comunicazione wireless

ProBee-ZE10

ProBee-ZS20S

CODICE MIP 2827159

**ProBee** ZigBee Certified

- Supporto completo della connettività wireless ZigBee 2007 / ZigBee Pro
- Coprocessore integrato 2.4GHz, IEEE 802.15.4 compliant
- Fino a 1.6Km (1 miglio) di portata tramite l'antenna dipolo opzionale 5 dBi
- Disponibilità di un Windows Tool per la semplice configurazione
- Supporta la configurazione remota e l'aggiornamento del firmware

E' un prodotto distribuito da elettroshop.com brilliant electronics since 1998

Per maggiori informazioni www.elettroshop.com oppure chiamare lo 02/66504794

SENA